

My students contributed in creating these lecture notes. I appreciate there efforts but I didn't reviewed them so they are neither guaranteed to be free from errors nor to be complete.

Hamid M.Gholizadeh

به نام هستي بخش گيتي

Internet Engineering

استاد: مهندس قلي زاده

تهيه و تنظيم:

گلناز تراب فام

مرکز آموزش عالی علمی - کاربردی جهاد دانشگاهی

انواع شبکه های ارتباطی:

شبکه های سوئیچینگ مداري.

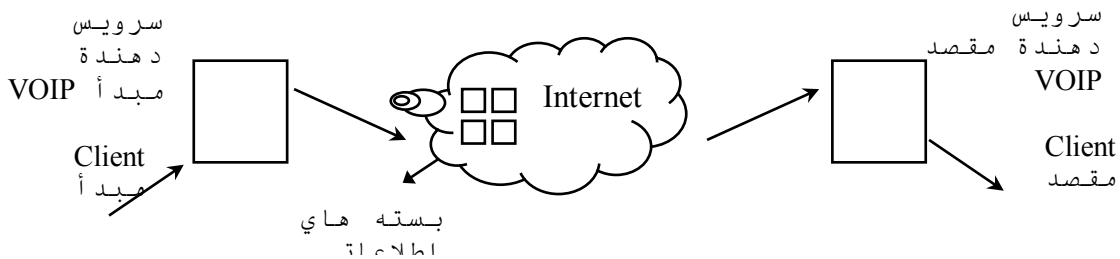
Circuit switched network

Packet switched

شبکه های سوئیچینگ بسته ای.
network

• اینترنت به صورت packet است.

سیستم ارتباطی VoIP:



انحصاری

سوئیچینگ مداري (CSN):
(delegated)

در شبکه های سوئیچینگ مداري در زمان ارتباط میان دو گره در شبکه تمامی مسیر اختصاص داده شده به این دو گروه به صورت انحصاری به این ارتباط تخصیص داده می شود و تا زمانی که این ارتباط برقرار است ارتباط دیگری نمی تواند از این کانال استفاده کند.

برای مثال اول باید درخواست ایجاد مسیر بدهیم. در تلفن برای ایجاد مسیر زنگ می زنیم موقعی که رنگ می زنیم سیگنال الکتریکی را به سوئیچ اولی می دهد به این سوئیچ که می رسد با سرعت نور حرکت کرده پس یک

موج الکترو مغناطیسی است. وقتی صفر و یک را به صورت موج ارسال می‌کنیم همان لحظه به آنجا نمی‌رسد بلکه با تأخیر می‌رسد و تأخیر ناشی از انتشارات راست- بعد به جلو نگاه می‌کند تا ببیند که شماره متعلق به مرکز خودش است یا به مرکز دیگری، یعنی پردازش می‌کند تا مسیر را پیدا کند با استفاده از شماره‌هایی که ما می‌فرستیم به جلو انتقال می‌دهد و دوباره با یک تأخیر ناشی از انتشار به دومی می‌رسد و آن هم با توجه به شماره شروع به زنگ زدن به طرف مقابل می‌کند. عملاً یک فرستنده به دو سوئیچ و یک گیرنده داریم. وقتی به گیرنده می‌رسد زنگ می‌خورد و گوشی را که طرف برداشت ارتباط برقرار می‌گردد. «صوت و تصویر فشرده نشده»

سوئیچینگ بسته‌ای (psn):

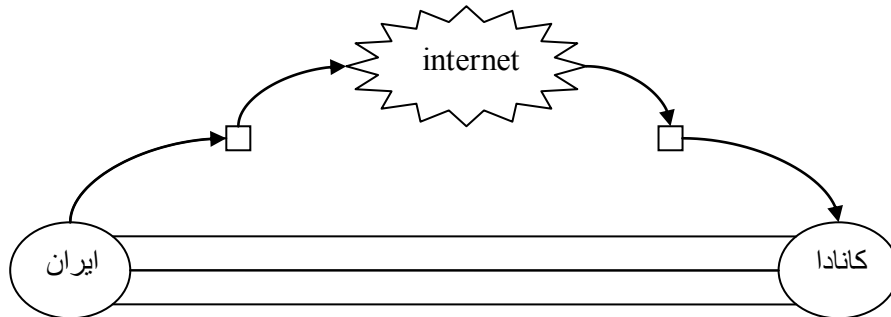
Sharing : داده ها تقسیم بندی می شوند

کانال ارتباطی میان دو گره در زمان برقراری ارتباط میان آنها به صورت اختصاصی در اختیار آن ارتباط قرار نگرفته بلکه می‌تواند میان سایر ارتباطات موجود یا بالقوه به اشتراک گذارده شود مثل شبکه های اینترنت یا شبکه های کامپیوتری .
{ دنیا به جزء های کوچک تقسیم می‌شود - paket از یک ایستگاه به ایستگاه دیگر انتقال پیدا می‌کند تا به مقصد برسد، برای اتصال دادن یک ترمینال به کامپیوتر و کامپیوتر به کامپیوتر دیگر استفاده می‌شود. «صوت و تصویر فشرده شده.» }

سوال : چرا استفاده از کارت های p2p (2 phone

phone) هزینه تماس مخابرات را کاهش

می دهد ؟



Analog → digital → service

شبکه های کامپیوتری:

مجموعه ای از سیستم های کامپیوتری متصل به هم که از طریق یک کانال ارتباطی با یکدیگر در ارتباط هستند .

هدف شبکه های کامپیوتری:

1) اشتراک منابع (سخت افزار ، نرم افزار)

2) تبادل داده

انواع شبکه های کامپیوتری از لحاظ گستردگی

جغرافیایی:

LAN (Local area network):

شبکه ای است خصوصی در ابعاد یک ساختمان که حداکثر ابعاد آن یکی دو کیلومتر است.

انتقال پخش (محیط مشترک) توپولوژی bus , حلقوی و توپولوژی ستاره ای از این نوعند.

کانال های ارتباطی : کابل های کواکسیال ، زوج سیم های به هم تابیده ، wireless (بی سیم)

wireless (بی سیم) : از طریق bluetooth و از طریق wi-Fi (802.11b) که هر دو امواج رادیویی هستند .

WAN (Wide area network) :

شبکه هایی در ابعاد یک شهر ، کشور ، کل جهان تک تک به هم وصل نمی شوند و اول باید کامپیوترها را به هم وصل کنیم و تشکیل شبکه LAN بدهیم و بعد شبکه های LAN را به هم متصل کنیم و تشکیل شبکه WAN دهیم .

کانال های ارتباطی : فیبر نوری ، بی سیم (امواج رادیویی) ، ماهواره ای

• سرعت انتقال در LAN بیشتر از WAN است.

Enterprise Network : به سازمان های بزرگ که در سراسر جهان شعب مختلف دارند مربوط می شود

انواع شبکه های WAN :

(value added network) van (1)

(Virtual Private network) vpn (2)

van (value added network) :

بعضی از شرکت ها مسئول نگه داری شبکه هستند و کسانی که به این شبکه نیاز دارند شبکه را از آن شرکت ها اجاره می کنند .

vpn (Virtual Private network) :

- مسیر ارتباطی بین دو نقطه در دسترس همه قرار دارد ممکن است بسته ارسالی به چندین نقطه برسد اما کسی که آدرس مقصد با آن (آدرس خودش) یکی است آن بسته را دریافت می کند .
- یک مسیر ارتباطی (شبکه) مجازی بر روی شبکه عمومی (internet) که ارتباط بین 2 نقطه به صورت کد شده و رمز شده برقرار است .
- برای بانک ها یک vpn در نظر گرفته می شود روی شبکه عمومی کشور که بستر امنی را برای بانک ها ایجاد می کند که از اطلاعات همدیگر نمی توانند با خبر شوند .

اینترنت :

یک شبکه جهانی گسترده است که از اتصال انواع شبکه های بزرگ و کوچک در سرتاسر جهان به همدیگر تشکیل شده است .

تاریخچه اینترنت :

در سال 1949 یک پروژه تحقیقاتی به نام ARPA در وزارت دفاع آمریکا شروع شد در شروع کار 4 تا گره (کامپیوتر) وجود داشت زمانی که شرکت های تجاری به آن ملحق شد شروع به رشد کرد از سال 1993 چیزی به نام internet به وجود آمد .

هیچ کس مسئول اینترنت نیست اینترنت یک شبکه عمومی است .

word wide web

وب (شبکه جهانی) :

به مستندي گفته مي شود که ترکیبی از متن و صوت و تصویر و یا ویدئو تشکیل شده باشد و هر کدام از این عناصر می توانند به صفحات وب دیگر ارجاع کنند .

Host : هر کام پیوتر متصل به شبکه را Host می گوئیم .

نکته : اینترنت فقط وب نیست ، وب جزئی از اینترنت است (وب = تار عنکبوت)

پروتکل (protocol) : یک سری قوانین و مقررات

مجموعه قوانین حاکم بر شبکه که امکان ارتباط میان دو کامپیوتر در شبکه را میسر می کند .
باید هر دو کامپیوتر آن پروتکل را بدانند .

مدل 7 لایه ای OSI برای پروتکل ها :

کاربرد

نمایش
جلسه
انتقال
شبکه
پیوند داده
لایه فیزیکی

مجموعه پروتکل (TCP/IP) :

پروتکلی که در شبکه اینترنت استفاده میشود پروتکل TCP/IP است
 پروتکل TCP/IP مجموعه ای از پروتکل
 هاست که IP و TCP
 نام دو پروتکل مشهور از این مجموعه
 است.

کاربرد
کاربردی
نمایش
جلسه
انتقال
شبکه
فیزیکی (پیوند داده فیزیکی)

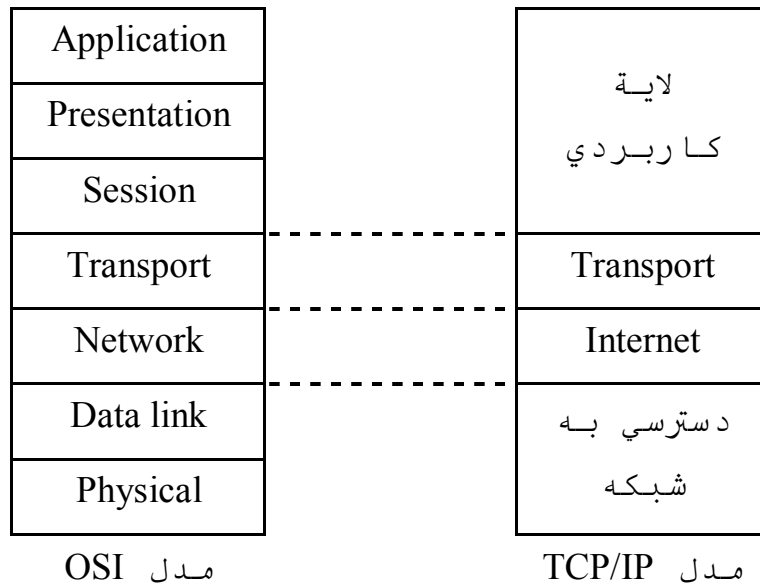
کاربردی : pop3 , SMTP , TelNet , HTTP , FTP

انتقال : UDP , TCP

شبکه : IP , ICP , ...

فيزيكي : 802.x (دسترسى به شبكه)

مقایسه دو مدل TCP/IP و OSI .



مزیت‌های OSI :

OSI یک سری مفاهیم را تعریف می‌کند که در TCP/IP تعریف نشده‌اند. پس در OSI مفاهیم از هم کاملاً مجزا هستند.

(1) سرویس (2) واسط (3) پروتکل

معایب OSI :

1. زمان‌بندی بد - تکنولوژی بد - اجرای بد - سیاست بد.

معایب TCP/IP :

TCP/IP در لایه سرویس, Interface و پروتکل را نمی‌توان تشخیص داد و واقعا یک مدل نیست, بلکه پشته‌ای از پروتکل است.

• تقسیم‌بندی لایه‌ها بر اساس سازمان دهی برای کنترل آن‌ها صورت گرفته.

شبکه نظام بانکی کشور را شبکه دیتا گویند ولی فقط بانکها از آن استفاده نمی‌کنند (فیبر نوری) بانکها از طریق خطوط UPN به هم متصل هستند.

پروتکل‌های مورد استفاده در لایه‌ها توسط اسنادی به نام RFC تعریف می‌شود.

RFC (Request for comment) :

RFC ها مستنداتی هستند در ارتباط با اینترنت و شبکه‌ها که شامل تحقیقات و یافته‌های جدید و روش‌های کاربردی در ارتباط با فناوری‌های مرتبط با اینترنت می‌باشند.

ویژگی‌های RFC :

1. مستندات غیر رسمی تری نسبت به استانداردها هستند.
 2. همه ی RFC ها به استاندارد تبدیل نمی شوند.
 3. هر RFC برای خود دارای یک شماره سریال یکتا است. مثل RFC 2026
 4. به صورت رایگان در دسترس عموم قرار دارد.
- IP u.6 → RFC 1883 Specification

: (Internet Engineering Task Force) IETF

مطرح می شود به عنوان استاندارد های اینترنت بر برخی از پیشنهادات که در قالب RFC می گزینند.

این سازمان در ارتباط با w3c (در رابطه با استاندارد های وب کار می کند) و ISO/IEC همکاری نزدیک دارد به عنوان مثال برای هر کدام از پروتکل های شبکه می توان RFC مربوطه را در اینترنت یافت هر RFC دارای یک شماره یکتاست به عنوان مثال RFC/1883 توصیف مشخصات مربوط به IP ورژن 6 می باشد .

: اینترنت 2

1. توسط 206 دانشگاه برای توسعه کاربرد علمی شبکه و تکنولوژی
2. با همکاری مشترک دانشگاه ، صنعت و دولت
3. تفاوت آن با اینترنت معمولی = پهنای باند زیاد (در حد گیگا)

4. زیر ساخت ارتباطی آن مجزا از شبکه اینترنت نیست.

- مزیت خطوط ADSL این است که از پهنای باند استفاده نشده خطوط مخابرات برای تبادل داده استفاده می کند و این کار توسط مودم انجام می دهد.

آدرس IP

آدرس IP یک چهار تایی به صورت a.b.c.d که هر کدام از این ارقام بین 0...255 هستند که هر کدام یک بایت است که کلاً 4 بایت می باشد. هر کامپیوتر متصل به شبکه باید دارای آدرس IP منحصر به فرد باشد.

$$(2^4)^4 = 2^{32} \times 4 = (10^3)^3 \times 4 = 10^9 \times 4$$

IP Address برای شبکه های داخلی :

آدرس IP شبکه های داخلی در این 3 رنج است :

172.16.a.a : (1) کلاس A

10.a.a.a : (2) کلاس B

192.168.a.a : (3) کلاس C

ارجاع به خود کامپیوتر در هر سیستم : 127.0.0.1

IP ورژن 6 :

با گسترش شبکه اینترنت از آنجا که با استفاده از IP ورژن 4 در تعداد کامپیوترهایی که می توانیم توسط این ورژن از آدرس IP آدرس دهی کنیم محدودیت وجود دارد IP ورژن جدید 6 معرفی شد.

a : a : a : a : a : a

a : عدد هگزادسیمال 2 بایتی

$$(2^{16})^8 = 2^{128}$$

2001:1db:0:0:0:0

مثال برای IP Address :

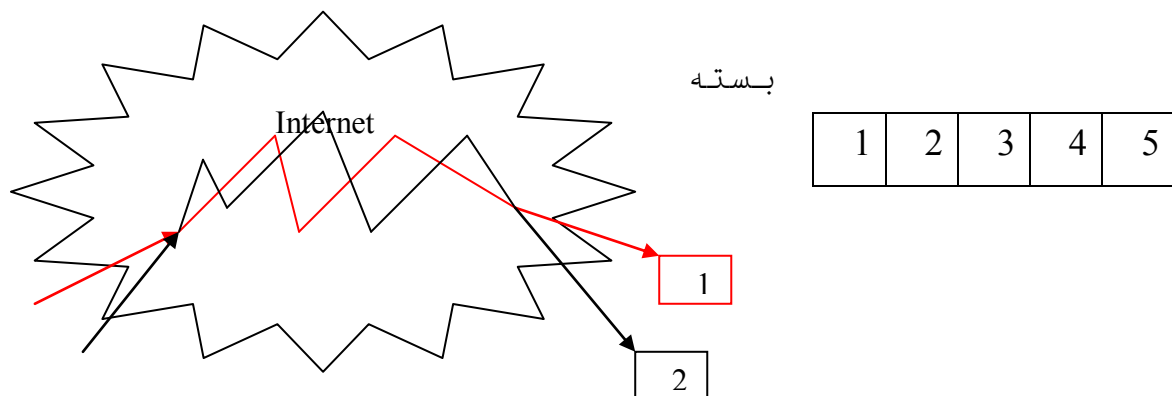
نکته : آدرس هایی که در IP ورژن 6 هستند با FE80 شروع می شوند .

مثال : 1 ای 128 0:0:0:0:0:1/128 ::1/128

- IP ورژن 5 برای هدایت ویدئو (Audio , video) در شبکه بوده و ماهیت IP ورژن 5 با 4 اصلا یکی نیست .

پروتکل IP¹ : در لایه ی شبکه است.

وظیفه پروتکل IP هدایت و مسیر یابی درست بسته های اطلاعاتی در شبکه است ترتیب درست بسته های دریافت شده در مقصد توسط این پروتکل تضمین نمی شود همچنین پروتکل IP تضمینی در مورد رسیدن تمامی بسته های ارسال شده ارائه نمی کند .

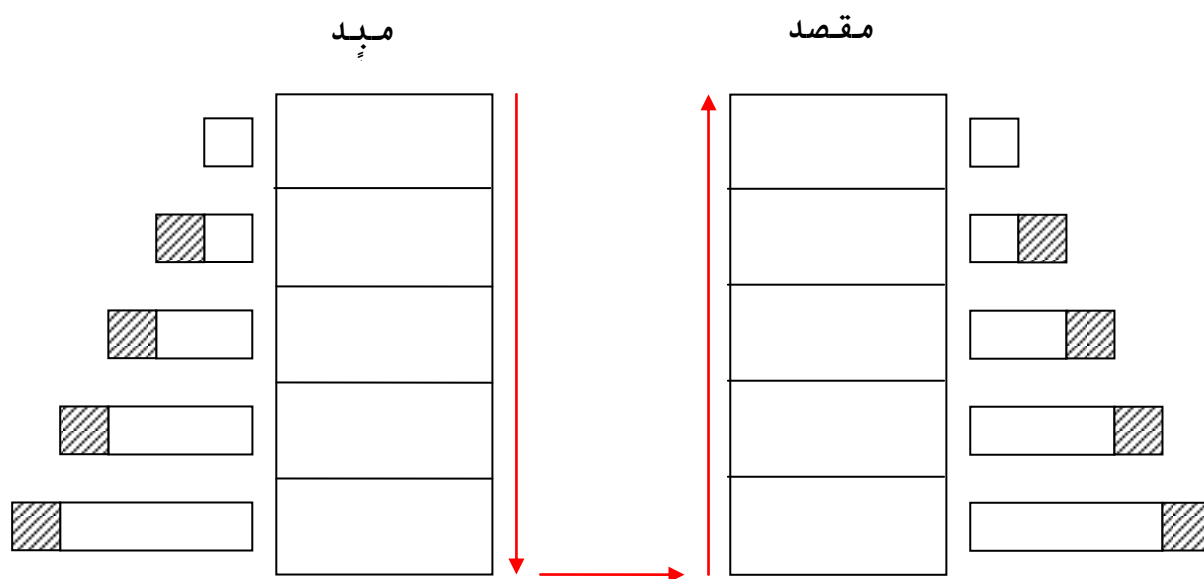


هر بسته اطلاعاتی در این پروتکل بصورت مستقل مسیر یابی می شود .

پروتکل TCP² : در لایه انتقال

1- Internet Protocol
2 - Transmission control protocol

نقطه ضعف هاي پروتکل IP توسط پروتکل TCP بر طرف مي شود پروتکل TCP بسته ها را در مقصد مرتب مي کند و تضمين مي کند که تمامی بسته ها به مقصد خواهد رسيد براي انجام اين کار لازم است که براي بسته هاي در يافتی توسط مقصد در اين پروتکل هنگام دریافت درست بسته (aknowledge) به مبدا ارسال شود .



ویژگی های پروتکل TCP :

1. داده ها را به صورت بسته های اطلاعاتی تکه تکه می کند.

2. گم شدن بسته‌ها را بررسی می‌کند.
3. وجود مشکل یا ایراد در بسته‌های ارسالی را بررسی می‌کند.
4. تقاضای ارسال مجدد برای بسته‌های گم شده یا خراب انجام می‌دهد.
5. بسته‌ها را در مقصد دوباره به صورت صحیح مرتب می‌کند.

پروتکل ¹UDP (User Datagram Protocol) :

این پروتکل عملکردی همانند پروتکل TCP دارد با این تفاوت که بررسی خطا در بسته‌ها انجام نمی‌شود بسته‌های گم شده را دوباره در خواست نمی‌کند و پیغام **acknowledge** نیز نمی‌دهد بنابراین این سریعتر از پروتکل TCP می‌باشد زمانی که در یک شبکه در انتقال سریع داده، انتقال کاملاً صحیح داده ارجحیت داشته باشد از این پروتکل استفاده می‌شود.

نکته: آدرس پورت مبدا و آدرس پورت مقصد جزء اطلاعات سرآیند (هدر) در پروتکل TCP می‌باشد.

آدرس port :

یک آدرس IP در شبکه یک کامپیوتر را بصورت یکتا مشخص می‌کند اما زمانی که داده ای به یک کامپیوتر

ارسال می شود برنامه های مختلفی در آن سیستم کامپیوتری ممکن است وجود داشته باشد .

نکته: آدرس port (شماره port) در یک سیستم کامپیوتری به صورت دقیق برنامه مورد نظر را مشخص می کند .

نکته: port 21 برای FTP است .

نکته: port 80 برای HTTP است .

نکته: یک برنامه به یک port خاص گوش می دهد ممکن است چندین برنامه همزمان به یک port گوش دهند .

مثال :

Start ⇨ run ⇨ cmd ⇨ netstate 192.168.1.3 : 5421

IPTable بعد از نصب firewall شماره port های قابل دسترسی را کنترل می کند اگر اجازه ارسال به port مربوطه را داشته باشد به آن port ارسال می کند و گرنه ارسال می کند .

Netstate -? : سوئیچ های netstate را می دهد .

IP config : آدرس IP و Subnet mask و DNS و default gateway را می دهد .

IP config/renew : یک IP آدرس جدید می دهد .

پروتکل FTP¹:

در لایه Application است.

پروتکل FTP جهت دریافت و ارسال فایل در محیط اینترنت مورد استفاده قرار می گیرد در این پروتکل بدون کدگذاری در محیط شبکه منتشر می شوند بنا بر

این در صورتی که داده ها در مسیر قطع (intercept) شوند می توانند توسط افراد نا خواسته مورد دسترسی قرار گیرند به همین دلیل از یک لایه امنیتی به همراه این پروتکل جهت کدگذاری داده ها استفاده می شوند پروتکل FTP که داده ها را به صورت کد شده منتقل می کند اصطلاحاً SFTP یا Secure FTP گفته می شود .

• برای برقراری یک اتصال FTP معمولاً به یک User name و Password نیاز است البته کاربر anonymous (نام مشخص - بی نام - ناشناخته) هم می توان برای اتصالات FTP تعریف کرد.

فرمت دستور FTP در cmd ویندوز ← آدرس سرور
ftp FTP

FTP 192.168.1.101

دستورات کاربردی در FTP

1. quit : برای خروج از برنامه ی FTP
2. dir : دایرکتوری جاری را لیست می کند.
3. ccl (remot directory) : به دایرکتوری مورد نظر تغییر مسیر
4. get (file name) : فایل ذکر شده را از FTP Server دریافت می کند.
5. put (file name) : فایل ذکر شده را از FTP Server منتقل می کند.
6. close : جلسه FTP ایجاد شده را خاتمه می دهد.
7. cd : تعویض دایرکتوری جاری

نکته: FTP فقط براي sharing کردن فایل هاست نمی توانیم تشخیص دهیم که چه کسی فرستاده است.
مثال:

Connected to 192.168.1.101

220 microsoft ftp service

User < 192.168.1.101<non>>: test

331 password requested for test

Password test

روش دیگر: در منوی آدرس my computer یا Internet explorer بنویسید:

ftp:// 192.168.1.101

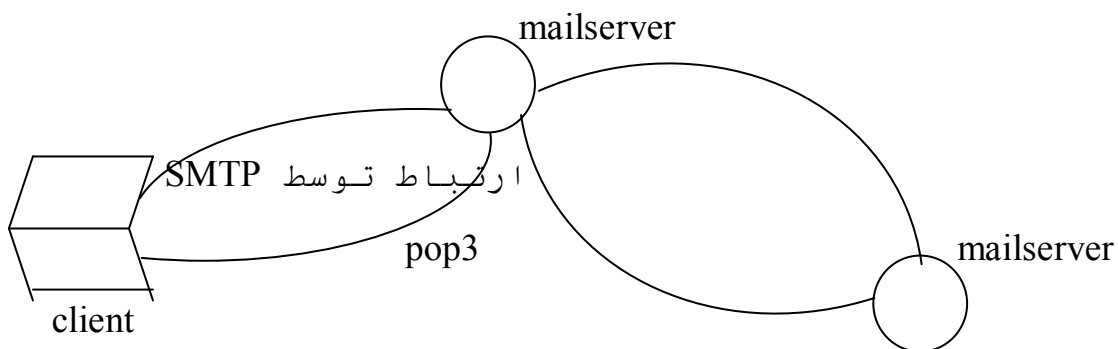
برای وصل شدن به شبکه:

(1) ftp:// 128.0.0.1

(2) نرم افزار vmware را نصب کنید

پروتکل SMTP¹:

برای انتقال پست های الکترونیکی (email) میان دو سرویس دهنده پست الکترونیکی استفاده میشود.



1- simple mail transfer protocol

آدرس سرویس

شماره پورت

telnet

SMTP دهنده

25

مثال:

telnet

SMTP.gmail.com

توسط این دستور می توانیم به سرور SMTP وصل شویم و telnet را برای چک کردن انواع پروتکل ها می توانیم استفاده کنیم .

25 پروتکل استاندارد SMTP است و می توانیم از پورت های متفاوتی استفاده کنیم که آن برای ایجاد امنیت می باشد . مثلا توسط این پروتکل در email خود می بینیم که افرادی non-valid برای ما email فرستادند .

عناصر یک پست الکترونیکی :

From : از طرف

To : آدرس گیرنده

CC : آدرس گیرندگان (Carbon copy)

BCC : آدرس هایی که دریک BCC قرار می گیرند

(blank CC)

Subject : عنوان

Body : بدنه email

نکته : در RFC 821 این پروتکل به صورت کامل تعریف شده است.

پروتکل Tel Net^۳ :

- وقتی با پروتکل Tel Net به یک remote Host ، login می کنیم انجام تمامی عملیات بر روی آن سیستم برای ما میسر می شود. به طوری که انگار به صورت فیزیکی پشت remote Host نشسته باشیم.
- در پروتکل Tel Net تعریف کاربر anony mous به خاطر مسائل امنیتی لحاظ نشده است به عبارتی باید برای login حتما باید user / pass داشت.
- ملاحظات امنیتی به خاطر ریسک پذیری بالای این پروتکل را باید مد نظر قرار داد.
- وقتی کسی Tel Net کند به شبکه مثل اینکه کل شبکه را در دست دارد و نشسته پشت همین شبکه.

نحوه دسترسی به Tel Net

Start > All programs > Accessories > communication > Hyper terminal

سپس باید تنظیمات را انجام دهیم.

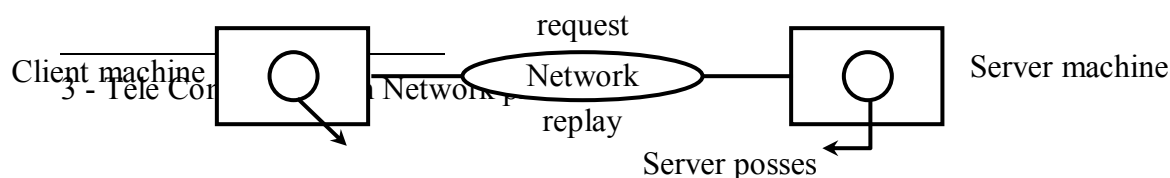
پروتکل های مرتبط با Email ← SMTP / POP3

پروتکل POP3 :

یکی از پروتکل هایی است که برای انتقال email ها از یک سرویس دهنده email به روی client استفاده می شود .

:IAMP (Internet Message Access Protocol)

همانند پروتکل POP3 عمل میکند .



پروتکل http^۴ :

پروتکلی برای انتقال صفحات وب در محیط اینترنت یا اینترنت می باشد .

نسخه جاری این پروتکل نسخه 1.1 است.

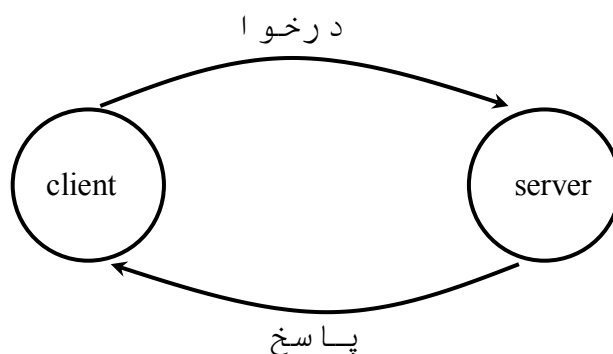
تفاوت نسخه 1.1 با 1 : تفاوت این نسخه با قبلی در

این است که در اتصال می تواند چندین صفحه را با

client مبادله کند و در قبلی برای هر صفحه که می

خواهیم بگیریم لازم بود که یک اتصال

انجام بدهیم .



نکته : خصوصیت بارز پروتکل http خاصیت state less بودن

آن است به این معنی که در این پروتکل سرور هیچ

گونه اطلاعی در مورد درخواست‌های قبلی که از یک client دریافت می‌کند نداشته و با درخواست جاری عیناً مانند یک درخواست اولیه برخورد می‌کند.

- Client : منظور از clientها در پروتکل http معمولاً همان web browserها مانند IE ، Netscape ، firefox ، ... است. لکن درخواست‌های HHP می‌تواند توسط هر نوع client دیگری نیز تولید شده و به سرور جهت گرفتن پاسخ ارسال گردد.

- Page : درخواست‌های ایجاد شده توسط client معمولاً برای دریافت یک Hyper media از سرور است.

- Hyper media : منظور از سندی است که محتوای متن ، تصویر و یا مولتی‌مدیا را در کنار هم می‌تواند داشته باشد و هر کدام از این عناصر می‌تواند به یک سند دیگر ارجاع داده باشد.

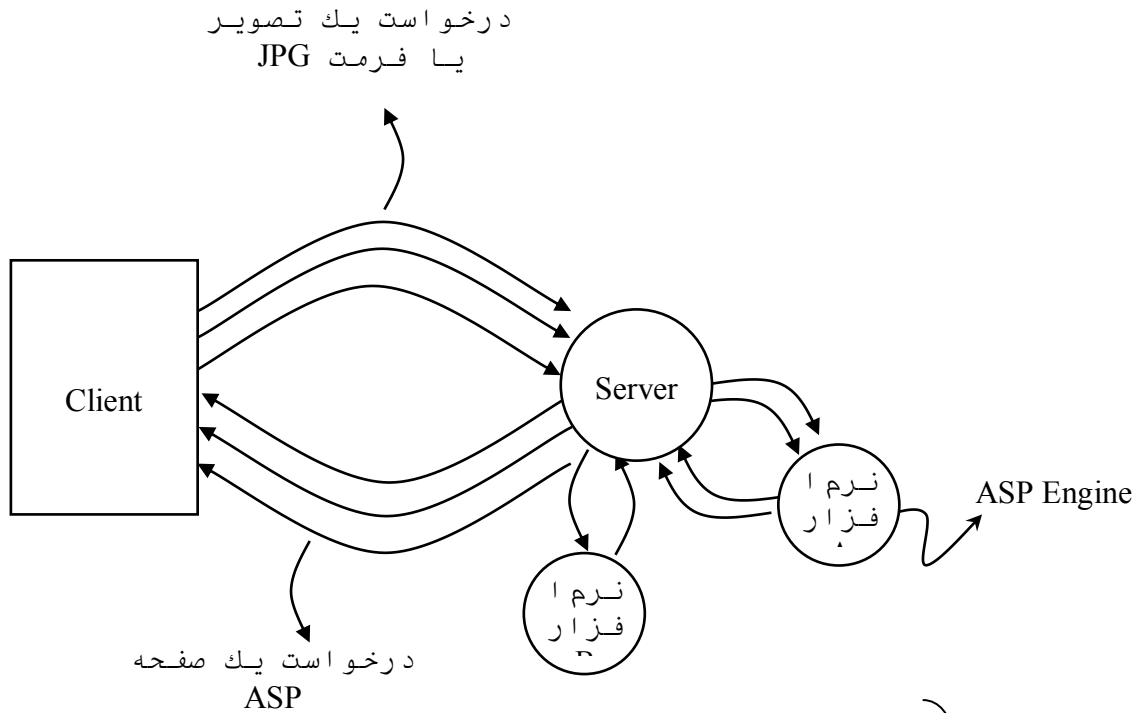
: Web server

وب سرور ها نرم افزار هایی در سمت server هستند که روی یک شبکه قرار می‌گیرند و وظیفه آنها پاسخگویی به درخواست های رسیده از سمت client ها برای دریافت صفحات وب است.

متناسب با هر درخواستی که به یک web server می‌رسد آن یا مستقیماً به درخواست پاسخ می‌دهد یا درخواست را برای تولید پاسخ مناسب به یک نرم افزار دیگر ارجاع می‌دهد (redirect می‌کند)

- در برخی موارد به سخت افزار سرور نیز عنوان وب سرور اطلاق می‌شود.

Web server : مدیریت و کنترل کننده است.



کاربردي : براي خدمت به

انسان به کار مي رود

نرم افزار :

سيستمي : مثل کامپایلر که

به صورت مستقیم کاربردي نيست

مثال براي وب سرور :

1) IIS°

توليد شرکت مايکروسافت ، قابل اجرا بر روي windows ، from .plat که يك نرم افزار openserver است و

قابل اجرا بر روی plat from های مختلف است و یا jakartatamcat که نرم افزار light است و open source .

آخرین نسخه آن تا به امروز IIS.6

نصب IIS :

Control pan : | Add or remote programs Add / remove windows

تیک مقابل گزینه IIS را می زنیم و آن را فعال می کنیم . components

دیدن سخت افزار IIS :

Control pan :| Administrator tools Internet Information server

Apache web server (2)

یک نرم افزار open source است و قابل اجرا بر روی platform های مختلف است.

Jakarta Tomcat (3)

آخرین نسخه : Tomcat 5

مفهوم https :

هنگامی که پروتکل http از لایه ssl استفاده می کند اصطلاحاً آن را https می نامند .

علت به وجود آمدن https این است که در پروتکل

http داده ها ب راحتی در اختیار افراد دیگر

نیز می توانند قرار بگیرد برای همین از پروتکل

https برای امنیت آن داده ها استفاده می شود

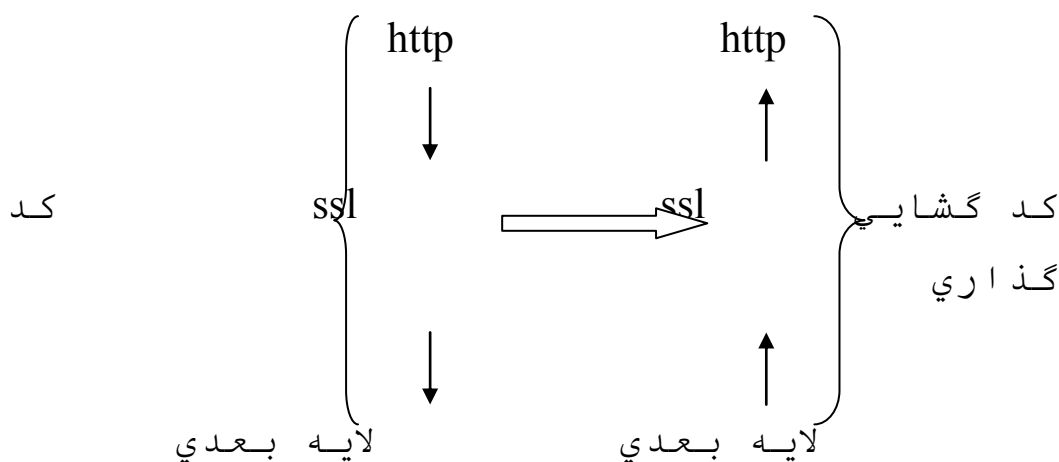
.

این پروتکل باعث می شود در پایین صفحه شکل قفل ظاهر شود که با **double click** کردن روی آن می توانیم اطلاعاتی را ببینیم که در آن قسمت **public key** (کلید عمومی) وجود دارد.

Ssl: کدهایی را که ارسال میشود را کدگذاری میکند که به صورت اتوماتیک انجام می شود.

کد گذاری داده ها توسط پروتکل **https**:

وقتی لایه **https** داده ای را ارسال می کند و به لایه پایین می دهد م وقتی مقصد آن را دریافت می کند وقتی به لایه **ssl** رسید آن کد را باز می کند و به لایه **http** می رود .



دستورات **http**

زمانی که یک اتصال **http** با سرور برقرار می شود **client** با استفاده از دستورات **http** درخواست خود را برای سرور ارسال می کند.

دستور **post / get**

این دستور برای درخواست یک صفحه وب از وب سرور ،
مورد استفاده قرار می گیرد

نکته : تفاوت دستور **get** با دستور **post** در این است
که در دستور **get** پارامترهایی را که می خواهیم برای
server ارسال کنیم به صورت رشته ای به انتهای آدرس
URL اضافه شده و به **server** ارسال می گردد در صورتی
که در دستور **post** این چنین پارامترهایی در قالب
هدرهای درخواست **http** به **server** ارسال می گردد.

دستور head :

این دستور مشابه دستور **get** است با این تفاوت که در
پاسخ ارسالی از سمت سرور تنها بخش **header** مربوط به
پاسخ به **client** ارسال می شود و از ارسال بدنه پاسخ
به **client** خودداری می کند.

دستور put :

برای ارسال یک صفحه به سمت وب سرور

دستور option :

در خواست متد هایی که وب سرور از آنها پشتیبانی می
کند .

دستورات ذکر شده ، دستورات پر کاربرد **http** است لکن
دستورات دیگری هم در این پروتکل تعریف شده است.

ساختار یک درخواست و پاسخ http :

درخواست **http** از 2 قسمت تشکیل شده است :

(1) **header** یا سرآیند

سرآیند شامل یک سری اطلاعات کمکی در ارتباط با درخواستها و پاسخهای رد و بدل شده در پروتکل http است.

(درخواست از چه browser و IP ای می آید و browser چه نوع encoding را پشتیبانی میکند.)

(2) body یا بدنه

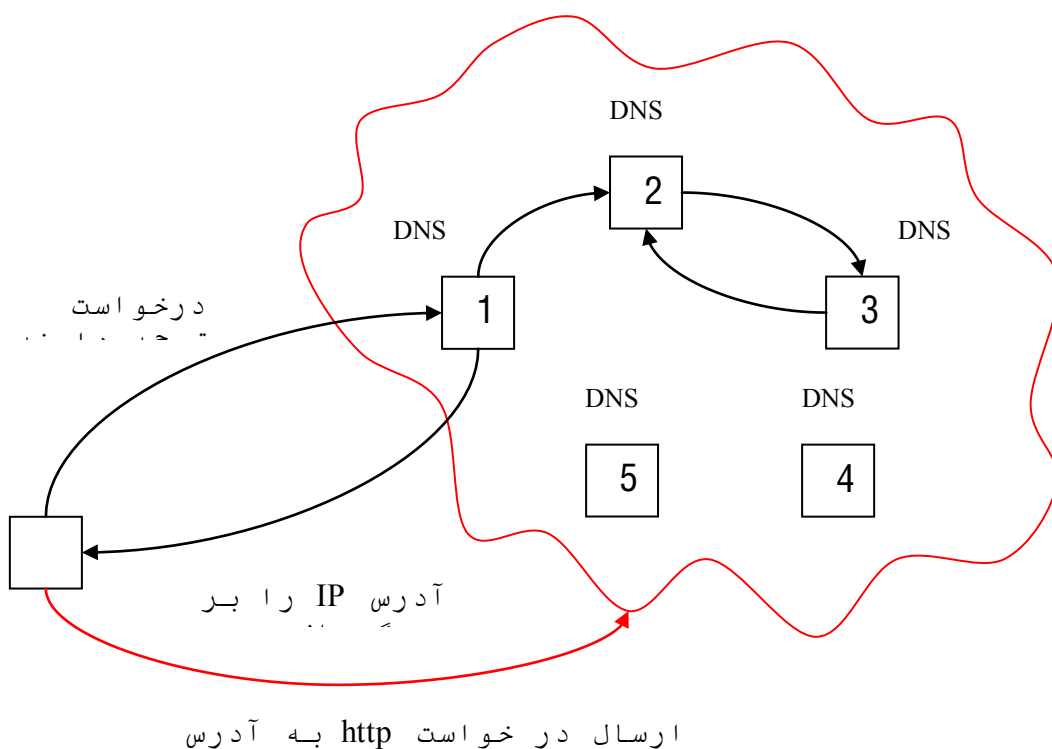
درخواستها معمولاً بدنه ندارند چون یک دستور است بدنه شامل محتوای منبع درخواست شده از سرور می باشد.

نکته: web browser ها در پشت صحنه درخواست http را تولید کرده و به server ارسال می کنند و پس از دریافت پاسخ قسمت بدنه پاسخ را از قسمت header جدا کرده و تنها آن را برای کاربر نمایش می دهند.

سیستم DNS^۱:

آدرس دامنه (Domain) : هر کامپیوتر متصل به شبکه در اینترنت دارای یک آدرس IP منحصر به فرد است. برای اینکه بتوانیم از منابع موجود در هر کدام از کامپیوترها استفاده نماییم باید آدرس IP آن کامپیوتر را حتماً بدانیم. به خاطر سپاری این آدرس IP برای کامپیوتر بسیار آسان ولی برای ما مشکل ساز است. برای همین دامنه ها معرفی شدند به عبارت دیگر دامنه ها که نام های با معنی هستند یک جانشین برای آدرس های IP به خاطر تسهیل به خاطر سپاری آنها است.

مثال : دامنه `yahoo.com` متعلق به شرکت `yahoo` به یک یا چند IP آدرس می تواند اشاره کند. در هر صورت برای درخواست یک سرویس از یک کامپیوتر متصل به شبکه ، در شبکه حتماً باید این درخواست به IP آدرس آن سرویس دهنده ارسال گردد. ما تنها دامنه آن سرویس دهنده را می شناسیم بنابراین به یک سرویس دهنده دیگر برای ترجمه دامنه ها به آدرس IP معادل آنها باید وجود داشته باشد. ما این سرویس دهنده را `DNS` یا `Domain name server` می نامیم.



وقتی که در خواست ترجمه دامنه به یک `DNS` رسید اگر آدرس IP مورد نظر در آن وجود داشته باشد پیغامی میفرستد و اگر نباشد به `DNS` مجاور خود می فرستد

واگر در DNS مجاود وجود داشته باشد جواب جستجو را به DNS مورد نظر فرستاده و آن DNS ، IP را به جدول خود اضافه مي کند .

ثبت دامنه :

ثبت يك دامنه در يكي از DNS هاي اصلي موجود در اينترنت است به طوري كه اين دامنه را به يك يا چند IP آدرس assign مي کند.

انواع آدرس هاي دامنه :

(1) آدرس هاي جهاني

(2) آدرس هاي مختص کشور ها

دامنه‌های پر کاربرد : عمومی

- مراکز اطلاعاتی ← .info
- مراکز نظامی ← .mil
- برای ارگانهای دولتی ← .gov
- مراکز تجاری ← .biz
- سازمانها ← .org
- مراکز آموزشی ← .edu
- مراکز تجاری ← .com
- سازمانهای مرتبط با شبکه ← .Net

دامنه‌های مختص کشورها

fr	فرانسه	uk	بریتانیا
		jp	ژاپن
us	آمریکا	ir	ایران
		ru	روسیه

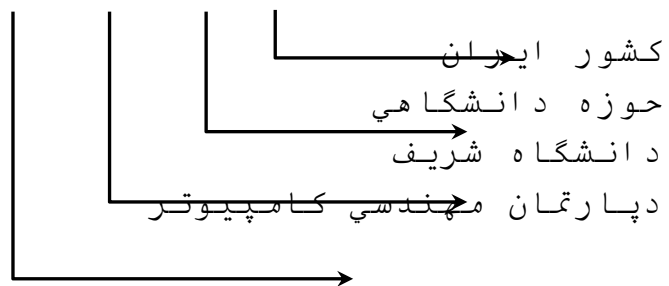
مثال :

ac.ir

sub domain

domain اصلی

CE. sharif. ac. ir



آدرس های مختص کشور ها :

.mil , .com , .net , .org , .inf , .edu , .bit , .gov

آدرس URL^۶ :

یک آدرس URL مشخص کننده یک منبع (فایل متنی ، تصویر ، مولتی مدیا) در یک شبکه (اینترنت) است.

ساختار یک آدرس URL :

آدرس منبع در سرور - آدرس دومین - پروتکل ارتباطی

مثال :

`http://www.Takfo.ir/DOC/page1.htm`

پروتکل آدرس منبع در سرور آدرس دامنه

ارتباطی

- صفحه وب

- تصویر

- مالتی مدیا

منبع :

نکته : آدرس IP 127.0.0.1 در هر کامپیوتر به

خود آن کامپیوتر اشاره می کند لذا اگر IIS بر روی

کامپیوتر نصب شده و در حال اجرا باشد با تایپ

آدرس زیر می توانیم از این وب سرور درخواست صفحه

یا منبع دیگر نماییم.

آدرس منبع `http://127.0.0.1/`

Local Host در ویندوز

وقتی آدرس URL یک صفحه در یک browser تایپ می شود

ابتدا آدرس Domain از داخل آن آدرس URL استخراج

شده و مطابق آنچه در قسمت DNSها توضیح داده شد

به آدرس IP ترجمه می شود و سپس یک اتصال http با

سرور مورد نظر که با همان IP آدرس مشخص می شود برقرار گشته و یک درخواست http برای منبع مشخص شده توسط آدرس URL به آن سرور ارسال می گردد اگر فرض کنیم درخواست برای یک صفحه html باشد سرور در پاسخ آن صفحه html را به browser ارسال می کند صفحات html معمولاً دارای تصویر و یا مولتی مدیا هستند browser کد داخل صفحه html را بررسی کرده و عناصری مانند تصویر و یا مولتی مدیا را به همراه آدرسهای URL آنها تشخیص می دهد و دوباره درخواستهایی برای دریافت عناصر مذکور مجدداً به سرور ارسال می کند و پس از دریافت پاسخها صفحه web را سازماندهی کرده و به کاربر نشان می دهد.

زبان HTML :

" زبان نشانه گذاری متن "

یک زبان نشانه گذاری متن است که به عنوان زبان طراحی صفحات وب مورد استفاده قرار می گیرد هر عنصر در داخل یک صفحه وب می تواند به یک صفحه وب دیگر ارجاع داده شود امروزه اکثر منابع موجود در اینترنت با استفاده از این زبان کد شده است.

ساختار زبان HTML

```
<html>
  <head>
    <title>          عنوان صفحه
  </head>
```

<body>

بدنه صفحه

</body>

</html>

نشانه‌هاي موجود در زبان html به صورت تگهاي باز و بسته است.

تگهاي موجود در زبان html محتويات صفحه web را نشانه گذاري مي کنند.

< نام tag > tag < /tag > محتوي

نکته : طبق استاندارد جديد html همه tagهاي جديد html بايد ساختار باز و بسته کردن tagها رعایت کنند در صورتی که tag اي داراي محتوا نباشد مي توانيم به جاي دو tag باز و بسته آن را به صورت زیر استفاده کنیم :

< نام tag / >

خصوصيات tagها (tag Attributs)

در داخل tagهاي زبان html مي توان متني به صورت نام / مقدار (name/values) تعريف کرد. اين خصوصيات براي تگهاي مختلف مي تواند متفاوت باشد و بيانگر خصوصيات اضافي در مورد tag مذکور باشد.

مي تواند با تگ کوتیشن جایگزین شود

>..... " مقدار خصوصیت " = نام خصوصیت نام

< tag

بدنه تگ

< نام tag >

تگ‌های زبان html :

< html > ... < /html >

تمامی صفحات html در داخل تگ html قرار می‌گیرند.

< head > ... < /head >

اطلاعات مربوط به سرآیند صفحه در این قسمت قرار می‌گیرد.

< body > ... < /body >

این قسمت حاوی بدنه اصلی صفحه web می‌باشد.

تگ‌های پر کاربرد در html :

< P > : برای ایجاد یک پاراگراف

< DIV > : برای ایجاد یک لایه

< B >, < I >, < U > : برای ایجاد حالت bold , italic , underline

< BR / > : برای پرش به یک خط جدید

< HR / > : برای رسم یک خط افقی

< TABLE > : برای ایجاد جدول

< FORM > : برای ایجاد فرم

< A > : برای ایجاد لینک

< IMG > : برای قرار دادن یک تصویر

< H1 > تا < H6 > : برای ایجاد متن‌ها به صورت سر
آیند

مثال : می‌خواهیم جدولی مطابق شکل زیر در زبان html
ایجاد کنیم معادل آن را در زبان html بنویسید.

1	2	3
4	5	6

```
< html >
  < head >
    < title > table  creation  < /title >
  < /head >
  < body >
    < table width= ' 300 ' border= ' 1 ' >
      < tr >
        < td > 1 < /td > < td > 2 < /td > < td > 3 < /td >
      < /tr >
      < tr >
        < td > 4 < /td > < td > 5 < /td > < td > 6 < /td >  < /tr >
    < /table >
  < /body >
< /html >
```

12		3
4	5	6

```
< html >
  < head >
    < title > table  creation  < /title >
  < /head >
  < body >
    < table  width= " 300 "  header= " 1 " >
      < tr >
        < td colspan= " 2 " > 12 < /td >    < td > 3 < /td >
```

```
</tr >
< tr >
< td > 4 </td > < td > 5 </td > < td > 6 </td >
</tr >
</table >
</body >
</html >
```

1	2	3
	4	5

```
< html >
< head >
< title > table creation </title >
< / head >
< body >
< table width= " 300 " border= " 1 " >
< tr >
< td rowspan= " 2 " > 1 </td >
< td > 2 </td > < td > 3 </td >
</tr >
< tr >
< td > 4 </td > < td > 5 </td >
< / tr >
</table >
</body >
</html >
```

نکته : tr سطر ، td ستون را مشخص می کند

فرم های web :

برای دریافت و جمع آوری اطلاعات از کاربران و ارسال آنها به سرور مورد استفاده قرار می گیرد برای ایجاد فرم های وب از تگ `< form >` استفاده می کنیم ساختار آن به صورت زیر است :

```
" Method = " مقدار " > /from >  
< from action = " مقدار
```

1 - خصوصیت Action :

مقدار نوشته شده در خصوصیت Action آدرس صفحه یا برنامه ای که مقادیر فیلدهای فرم بایستی به آن ارسال شود را مشخص می کند.

2 - خصوصیت Method :

مقدار این خصوصیت Get یا Post است و نحوه ارسال مقادیر فیلدهای فرم به آدرس مشخص شده در Action را مشخص می کند.

در حالت Get :

در این حالت مقادیر فیلدهای فرم ها از طریق اضافه شدن به انتهای آدرس URL مشخص شده در قسمت Action به Server ارسال می شود به صورت زیر :
`{http : // Domain Name. com/login.asp}?user= Ali & pass=123`
متن شامل نام / مقدار فیلدهای فرم Login

فرم login متناظر :

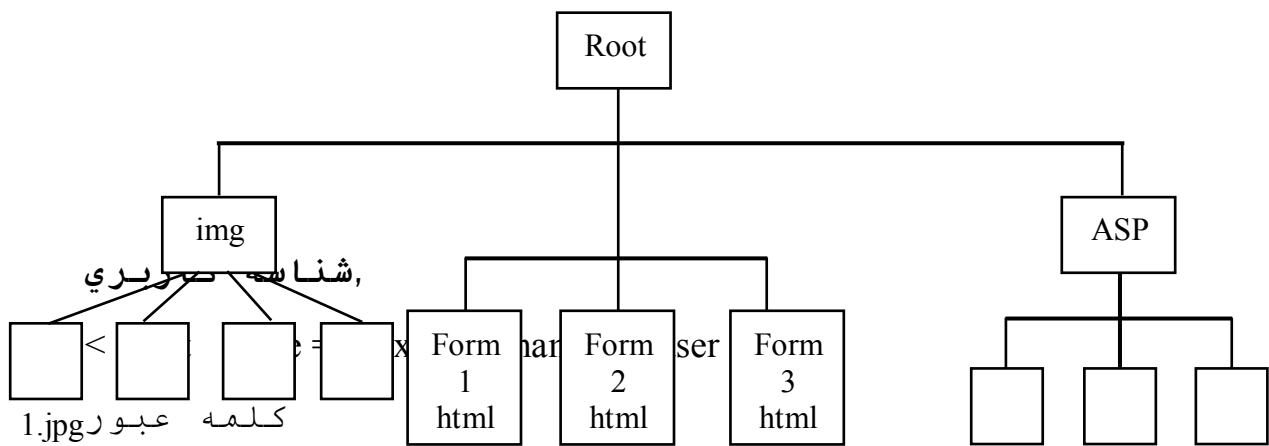
.....

< form Action= " http://Domain name.com/login.asp " Method= " Get " >

تفاوت آدرس نسبي و كامل با مثال :

< img href= " img/1.jpg " > آدرس نسبي

< img href = " http ://my
Domain.com/img17.jpg " > آدرس كامل



< input type = " password " name = " pass " >

در حالت post :

در حالت post پارامترهاي فرم در داخل header درخواست http به سرور ارسال مي شود و چون در قالب header به سرور ارسال مي شود ما آن پارامترها را نمي بينيم .

عناصر داخل فرم :

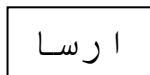
< input type = "text" , value = " مقدار اوليه " , نام = " name = "

در مقابل value هر چه بنويسيم در جعبه آن را نشان مي دهد



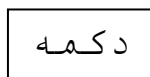
< input type = “submit ” , name = نام “ , value = “ ارسال ” />
“

باعث ایجاد دکمه ارسال به شکل زیر خواهد شد



< input type = “ button ” , name = نام “ , value = “ دکمه ” />
“

باعث ایجاد دکمه به شکل زیر خواهد شد ولي مثل بالايي
باعث ارسال نمي شود .



< input type = “ password ” , name نام “ , value = “ دکمه ” />
= “



< input type = “ hidden ” , name = نام “ , value = “ دکمه ” />
“

چيزي نشان نمي دهد براي اين است که پارامتر هاي مخفي
در صفحه بگذاريم و به درد برنامه نويسي مي خورد .

< input type = “ radio ” , نام “ , value = “ true “ , name = “a1” />
name = “

< input type = “ radio ” , نام “ , value = “ false “ , name = “a1” />
name = “



اگر نام آنها یکی باشد فقط یکی از آنها را می توانیم انتخاب کنیم ولی اگر متفاوت باشد هر دو را می توان انتخاب کرد

```
<input type="checkbox" name="a1" value="false" />
```

باعث ایجاد شکل می شود .

مثال :

```
<select name="sport" size="10">
<option value="kardani">kardani </option>
<option value="karshenasi">karshenasi </option>
<option value="arshad">arshad </option>
<option value="phd">phd </option>
</select>
```

کار دانی	>
----------	---

در اینجا هر مقطعی را انتخاب کنیم و دکمه submit را بزنیم باعث می شود magta = kardani شود و ارسال شود.

نکته : اگر در اینجا size = 4 شود combo box تبدیل می شود .

Kardani Karshenasi Arshad phd
--

```
<textarea name="favorites">
```

بدنه textarea	▲▼
---------------	----

دستورات

</form>

نکته : وقتی روی دکمه ارسال کلیک می‌کنیم باعث می‌شود اطلاعات فرم به قسمت p1.html که در قسمت action مشخص کردیم برود .

نکته : در متد get داده های ارسالی به صفحه مقصد بصورت مقادیر مقدار / نام در فرم شخص می‌شود .

<form action = “ p1.html “ , method = “post” , name = “form1” >

دستورات

</form>

در این حالت باعث می‌شود اطلاعات ارسال شود ولی نام و مقدار آنها در قسمت آدرس نشان داده شود

سر

بدنه کد
html

انواع شیوه های برنامه نویسی در وب :

1- برنامه نویسی سمت client

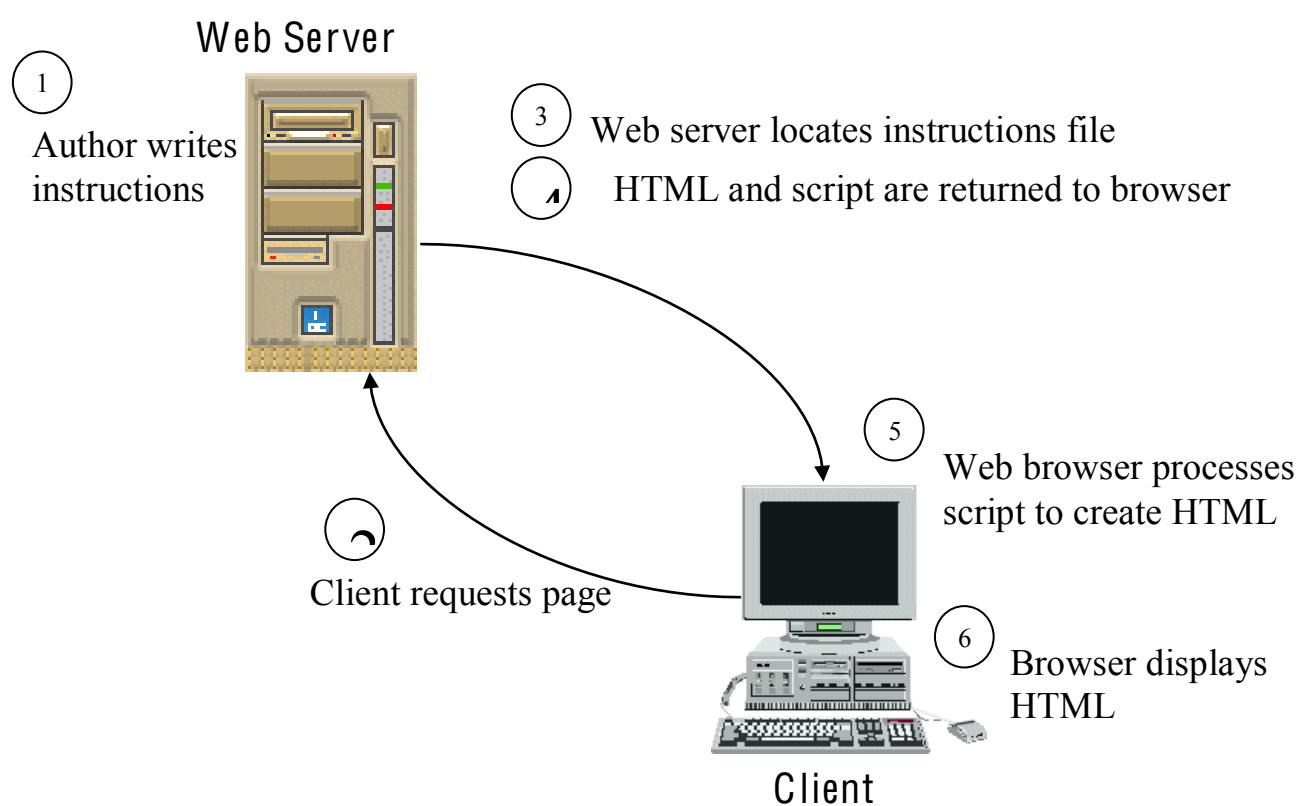
2- برنامه نویسی سمت server

3- برنامه نویسی ترکیبی

صفحات پویا : صفحاتی هستند که بر اساس عملکرد کاربر و شرایط جاری و حاکم بر زمان تعامل با آن صفحات ایجاد می‌شوند و این امکان را می‌دهند که آنها را برای هر کاربر سفارشی نماییم .

مدل برنامه نویسی سمت client :

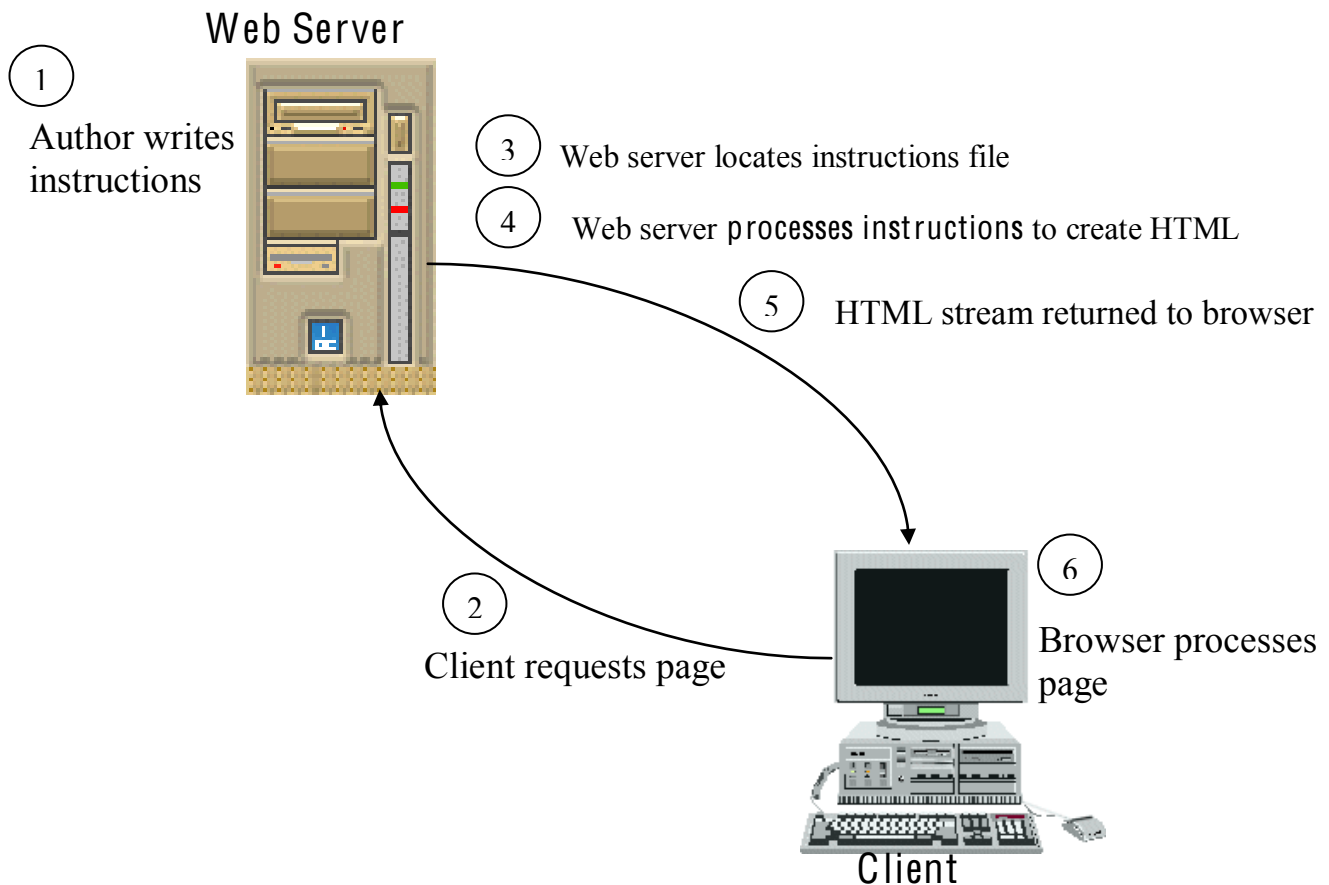
در برنامه نویسی سمت client پس از اینکه درخواستی مبني بر دریافت یک صفحه از سمت client به سرور ارسال می شود سرور صفحه مربوطه را به همراه کدهای برنامه نویسی به سمت client ارسال می کند و client این محتویات را گرفته و با اجرای کدهای همراه آن صفحه نهایی ایجاد شده را به کاربر نشان می دهد.



1. یک صفحه وب HTML به همراه کدهای سمت client نوشته شده بر روی سرور قرار می‌گیرد.
2. client صفحه مورد نظر را از سرور درخواست می‌کند.
3. سرور صفحه مورد درخواست client را به همراه کدهای مربوط به آن صفحه جستجو می‌کند.
4. صفحه HTML به همراه کدهای آن به client (همان browser) ارسال می‌شود.
5. client کدهای ارسالی را پردازش کرده و با استفاده از صفحه HTML ارسالی از سمت سرور صفحه HTML نهایی را تولید می‌کند.
6. صفحه نمایش داده می‌شود.

مدل برنامه نویسی سمت سرور^۱

در برنامه نویسی سمت سرور هنگامی که از طرف client درخواستی سمت سرور ارسال می‌شود سرور مطابق این درخواست صفحه مورد نظر را به همراه کدهای آن صفحه پیدا کرده پس از پردازش کدها صفحه نهایی را تولید کرده (ایجاد کرده) و آن را برای client ارسال می‌کند و client این صفحه را گرفته و بدون هیچ پردازشی بر روی کدها آن را نمایش می‌دهد.



1. صفحه HTML به همراه کدهای سمت سرور ایجاد شده بر روی سرور قرار می گیرد.
2. client صفحه را از سرور درخواست می کند.
3. سرور صفحه و کدهای درخواستی را جستجو می کند.
4. سرور کدهای درخواستی را پردازش کرده و با استفاده از html صفحه نهایی را تولید می نماید.
5. صفحه تولید شده به client ارسال می شود.

6. صفحه تولید شده توسط client نمایش می یابد و به کاربر نشان می دهد.

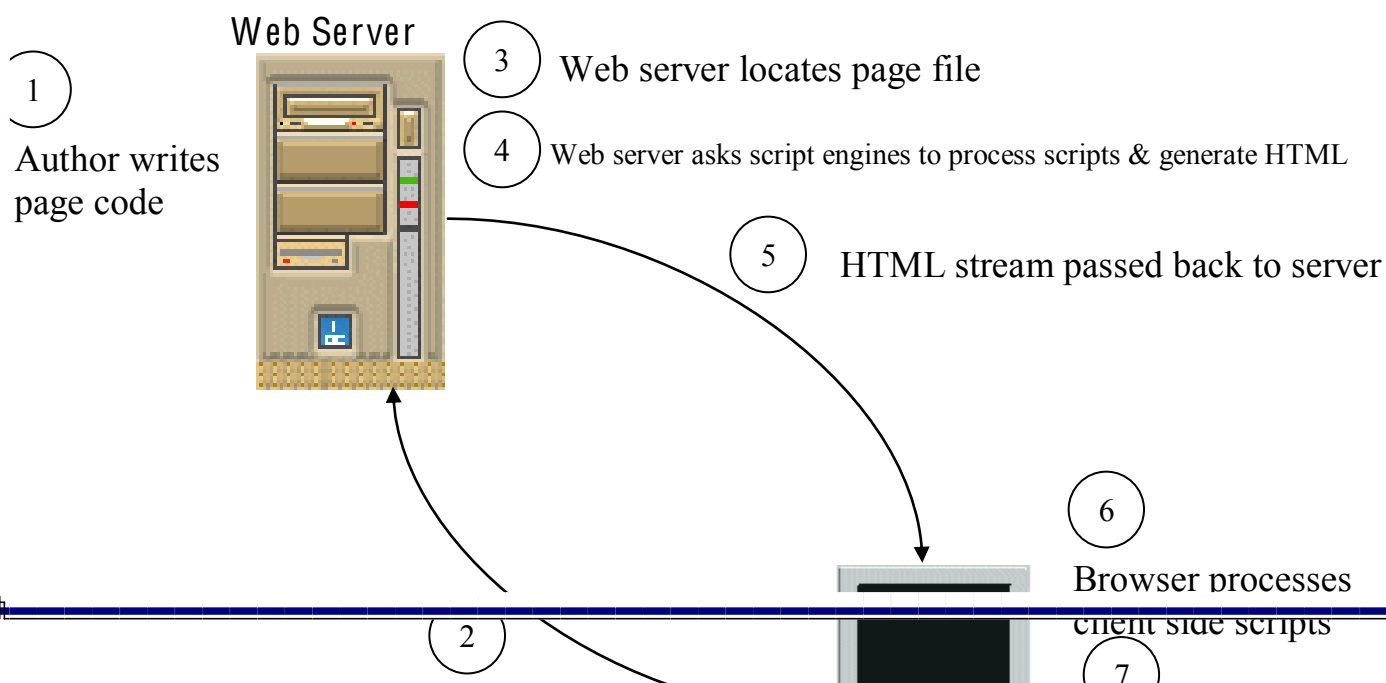
مقایسه برنامه نویسی سمت سرور با برنامه نویسی سمت client :

تفاوت اصلی آنها در محل پردازش کدهای مرتبط شده با صفحات html است. در برنامه نویسی سمت سرور کدهای برنامه نویسی مربوط به صفحات در سمت سرور مورد پردازش قرار می گیرند در حالی که در برنامه ریزی سمت client این کدها در سمت client پردازش می شود.

نکته : باتوجه به مقایسه مبنایی اخیر می توان مزایا و معایب هر 2 نوع برنامه نویسی را مورد تجزیه و تحلیل قرار داد.

مدل برنامه نویسی ترکیبی :

در برنامه نویسی ترکیبی ما از ترکیب 2 حالت اخیر استفاده می کنیم بدین معنی که برای صفحه ی html مربوطه کدهای سمت سروری وجود دارد که در سمت سرور مورد پردازش قرار می گیرد و هم چنین کدهای سمت client ای وجود دارد که در سمت client پردازش می شوند.



1. صفحه HTML به همراه کدهای سمت سرور و کدهای سمت client ایجاد شده بر روی سرور قرار می گیرد.
 2. صفحه مورد نظر توسط client از سرور درخواست می شود.
 3. سرور صفحه درخواستی را به همراه کدهای سمت سرور کدهای سمت client مورد جستجو قرار می دهد.
 4. سرور کدهای سمت سرور مربوط به صفحه html درخواستی را پردازش کرده.
 5. خروجی تولید شده را به همراه کدهای سمت client به client ارسال می کند.
 6. client کدهای سمت client را بر روی صفحه html ارسال شده از سمت سرور اجرا می کند.
 7. خروجی نهایی تولید شده نمایش داده می شود.
- تکنولوژی‌های مورد استفاده در برنامه نویسی سمت client :**

برای برنامه نویسی سمت client معمولا از زبانهای اسکریپتی استفاده می شود. کدهای نوشته شده توسط این زبان ها با کدهای html مربوطه به صفحه ترکیب

شده و معمولا این کدها در داخل همان صفحه html قرار می گیرد.

نکته : زبان های اسکریپتی در مقایسه با زبان های کامل مثل java ، VB.net و . . . امکانات کمتری برای برنامه نویسی در اختیار ما قرار می دهند این ویژگی باعث می شود که منابع مورد نیاز برای اجرای کدهای نوشته شده توسط زبان های اسکریپتی کمتر از منابع مورد درخواست در برنامه های نوشته شده توسط زبان های کامل باشد.

زبان های برنامه نویسی سمت client

1. java script
 2. VB script
 3. java Applets
- با استفاده از زبان java

java script

زبان java script یک زبان اسکریپتی است که برای اولین بار توسط Net scape با عنوان live script معرفی شد. Net scape این زبان را برای نوشتن برنامه های سمت سرور و client خود استفاده می کرد با مشارکت شرکت sun این زبان به java script تغییر نام داده و برای نوشتن کدهای سمت client مورد استفاده قرار می گیرد.

بر خلاف نام java script این زبان هیچ ارتباطی با زبان java ندارد و فقط در نحو syntax تشابهات زیادی با java دارد.

- یک برنامه کامل ارائه نمی دهد (چون محدود است) و مستقل از بستری است که browser روی آن نصب شده یعنی فرق نمی کند windows باشد یا unix و...

ویژگی های زبان java script

1. کدهای نوشته شده در سمت client اجرا می شود نه در سمت سرور
2. برای اجرا تنها نیاز به یک web browser دارد.
3. یک زبان case sensitive است.

حساس به بزرگ و کوچک بودن حروف

4. توسط تقریباً تمامی browserها پشتیبانی می شود.
- نکته:** زبان های VB.script ، Jscript نیز توسط مایکروسافت جهت کد نویسی سمت client ایجاد شده اند. زبان VB.script دارای syntax (شکل و شمایل زبان) ای مثل زبان ویژوال بیسیک دارد و jscript هم تقریباً مشابه زبان java script است.

• VB.script تنها توسط J.E پشتیبانی می شود.

4. برای اجرا حتماً نیاز مند یک browser است و به تنهایی نمی تواند اجرا شود.

مقایسه java script با زبان java

1. java یک زبان برنامه نویسی کامل است در حالی که java script یک زبان برنامه نویسی script ای است.

2. قدرت برنامه نویسی با استفاده از java بیشتر از java script است.

3. برنامه های نوشته شده توسط java عمدتاً برای اجرا نیاز به browser نداریم.

4. برنامه های نوشته شده توسط java برای اجرا بایستی compile شوند.

5. شباهت میان java ، java script تنها در نام آنها و syntax (نحوه) برنامه نویسی در آنها است. (ساختارها و قوانین برنامه نویسی)

کاربردهای زبان java script :

1. برای بررسی اعتبار فرم های وب

زمانی که کاربر در یک صفحه وب فرمی را پر کرده و با زدن دکمه درخواست را ارسال اطلاعات به سمت سرور می شود قبل از اینکه این اطلاعات به سمت سرور ارسال شود توسط کدهای java script می توانیم تا حدودی درستی و صحت اطلاعات وارد شده را بررسی نموده و در صورت لزوم پیغام های مناسبی برای کاربر تولید نماییم.

2. ایجاد تصاویر و متون متحرک یا انیمیشن دار

3. براي کنترل و تعامل با فرمهاي html و توليد

پوياي كد هاي html

با استفاده از فرمهاي html قادر خواهيم بود در يك پنجره browser بيش يك صفحه web بار گذاري نماييم .

4. كار با كوكيها در سمت client

5. پاسخگويي به رويدادهاي صفحات وب :

با استفاده از اين زبان مي توانيم به رويداد هاي كه در يك صفحه وب اتفاق مي افتد مثل حركت داده ماوس روه عنصر صفحه وب ، كليك كردن ماوس ، فشار دادن دكمه كيبرد ، load شدن يك صفحه وب و
عكسالعمل نشان دهيم .

: js script

این زبان در سال 1995 توسط شرکت ماکروسافت معرفی شد و برای اولین بار در `internet explorer` ورژن 3 پشتیبانی شد این زبان تقریباً شبیه زبان

java script است

ابزارهاي مورد نیاز برای نوشتن و اجرای کدهای java
: script

1. یک web browser J.E ، fire fox ، net scape ، opera
و...

2. یک ویرایشگر متن text editor ، dream viewer ،
note pad ، frontpage و...

نحوه استفاده از زبان java script در صفحات وب :

1. قرار دادن کدهای java script در بین 2 تگ </script >
<script > در داخل صفحه به صورت زیر :

```
<script language = "java script 1.5 ">
```

```
Document.write ( " < h1> hello world < /h1> " );
```

```
< /script >
```

- در این حالت کد هم می تواند در قسمت head و هم در قسمت body قرار گیرد.

- توابع java script بهتر است در قسمت head صفحه قرار گیرد تا هنگام استفاده از آنها در قسمت body از بارگذاری آنها اطمینان حاصل گردد. اگر تابعی در قسمت body قرار گیرد بلافاصله پس از بارگذاری صفحه اجرا خواهد شد.

2. قرار دادن کدهای java script در یک فایل جداگانه با پسوند .js و سپس استفاده از آن فایل با ارجاع از داخل صفحه وب به صورت زیر :

```
<script language = "javascript 1.5" src = " my java script file. js " type = " text/javascript" >< /script >
```

اینجا چیزی نمی نویسیم بلکه یک فایل را فراخوانی می کنیم .

قرار دادن کدهای `java script` در یک فایل جداگانه این امکان را به ما می دهد که از کدهای نوشته شده در بیش از یک صفحه `html` استفاده نماییم . دقت کنید که در داخل فایل `myjava script file. js` دیگر از تگ `script` استفاده نمی کنیم فقط قسمت بالا را می نویسیم و می توانیم در `head` یا `body` به کار ببریم .

3. قرار دادن کدهای `java script` به عنوان مقادیر خصوصیات عناصر فرم های وب

```
<body>
```

```
<p> <input type =”button” value=”click me” onclick=”javascript” :  
alert (‘this is athrid way of coding gava script’) />
```

وقتی از این روش استفاده می کنیم که کد `javascript` ما یک خطی باشد .

نحو زبان java script :

ساختار برنامه های `java script` بسیار شبیه زبان `C` می باشد زبان `java script` در بسیاری از موارد مانند تعریف متغیر ها ، تعریف خود متغیر ها تخصیص مقدار متغیر ها از انواع مختلف به همدیگر ، پارامتر بازگشتی توابع و... زیاد سختگیر نیست .

مفاهیم اولیه زبان java script

داده های اولیه `princitive data types`

1. `number` برای ذخیره سازی

اعداد (صحیح ، اعشاری)

2. string
رشته‌ها
برای ذخیره سازی

3. Boolean
مقادیر True ، False
برای ذخیره سازی

تعریف متغیر

برای تعریف متغیرها در زبان
کلیدی var استفاده می‌کنیم.
java script از کلمه

var $\pi = 3.1416$, X , Y , $name = 'Dr . Dave '$

قوانین

1. نام متغیرها با حرف یا
under score (-) باید
شروع شود.

2. اسم متغیرها حساس به حروف بزرگ و کوچک است.

3. هر متغیری که تعریف می‌شود می‌تواند هر نوع
داده ای را در داخل خود ذخیره کند.

4. متغیرهایی که در داخل یک تابع تعریف می‌شوند
تنها از داخل آن تابع قابل دسترسی هستند.

5. متغیرهایی که خارج از یک تابع تعریف می‌شوند
متغیرهای سراسری نامیده می‌شوند و از داخل همه

توابع قابل دسترسی هستند $Var K = 1.41$

با وجود اینکه در بالا $number$ است ولی $java script$ سخت K
 $= "book "$

نمی‌گیرد و اینها را قبول می‌کند و با وجود اینکه
نام متغیر $K = True$

در همگی یکی است. $K = 1.74$

عملگرها :

برای انجام عملیات بر روی یک یا دو متغیر مورد استفاده قرار می‌گیرد.

1. عملگرهای ریاضی ++ -- + - / * %
2. عملگرهای مقایسه ای < <= == > >= !=
3. عملگرهای منطقی
4. عملگرهای انتخابی += -= *= /= %=

$A += 4 = A = A + 4$ $A -= 4 = A = A - 4$

$A *= 4 = A = A * 4$ $A /= 4 = A = A / 4$

$A \% 4 =$ باقی مانده تقسیم 4 بر A
 $A = A \% 4$

توضیحات

برای افزودن توضیحات به کد java script به دو طریق می‌توانیم عمل بکنیم :

1. خطی که به اول آن `||` اضافه شود به عنوان توضیح محسوب خواهد شد.

- فقط برای یک سطر توضیح ← اگر بیش از یک سطر باشد در ابتدای هر سطر `||` بگذاریم.
||.....
||.....

2. خطوطی که بین دو علامت `/*` و `*/` قرار می‌گیرد به عنوان توضیح در نظر گرفته می‌شود.

- برای چند سطر توضیح
- در پاسکال ← { }

عبارتها statements

1. عبارات ساده : مانند انتساب یک تعداد به

یک متغیر $p = 1.41 + 2;$

2. عبارتهای ترکیبی : آنها در داخل {...} {

قرار می گیرد.

```
{ statement 1;
```

```
Statement 2;          { p = 2 + 1.41;
```

```
                        K = p + 4;
```

```
                        S = " Ali ";
```

```
Statement n }          }
```

نکته : در انتهای تمامی عبارتها به جز آخرین عبارت

در زبان java script از استفاده می کنیم.

- اگر دستور یکی باشد لزومی ندارد داخل {} قرار دهیم.

3. عبارت شرطی if / else if (name ==

```
" Ali " ) {
```

```
M = " you are Ali " ;
```

```
Flag = True; }
```

- شرط همواره داخل () قرار می گیرد در else { M = " you are not Ali ";

تمامی statement ها این گونه است. flag = false; }

می توان else را حذف کرد چون اگر شرط برقرار باشد وارد می شود اگر نه بعد از شرط را انجام می دهد.

4. عبارت while (flag == True) {

یک عبارت ترکیبی

تا زمانی که شرط while درست باشد عبارت داخل while تکرار می شود.

5. عبارت for برای ایجاد حلقه تکرار عبارتی که پس از یک بار تکرار اجرا می شود

For (i = 1 ; i <= 100 ; i++) {

Sum = 55 ⇒ خروجی : sum += 1 شرط اتمام

حلقه مقدار دهی اولیه به متغیر شرطی

آرایه‌ها

به سه طریق می‌توانیم آرایه‌ها تعریف کنیم
یک آرایه از نوع رشته کد حاوی سه عنصر است

1) `var colors = [" red " , " green " , " blue "];`

یک آرایه بدون مقدار دهی اولیه یا طول اولیه
تعریف می‌کند

2) `var colors = new array ();`

3) `var colors = new array (3);` یک آرایه به طول 3 تعریف می‌کند .
`array (s);`

`var colors = new array ("red" , "green" , "blue");` دقیقاً مانند 1 عمل می‌کند .

نکته : آرایه در `java script` از عنصر صفر شروع می‌شود
همانند `c` و `java` بر خلاف پاسکال که از 1 شروع می‌شود
.

6. عبارت with

در داخل عبارت که به صورت زیر نوشته شود :

```
With ( document . my form ) {
```

```
    } عبارت داخل with
```

یعنی برو در `document . my form` مقدار `result . value` را با `myinput . value` جایگزین کن .

```
• With ( document . my form ) {  
    Result . value = compute ( myinput . value )  
}
```

هر دو معادل است

```
• Document . myform . result value =  
    Compute ( document . Myform . myinput . value );
```

توابع ۱۱ :

توابع و پروسیجرها هر دو در زبان جاوا اسکریپت با کلمه کلیدی Function تعریف می شوند .

- اگر یک تابع دارای مقدار بازگشتی باشد با استفاده از کلمه کلیدی returned مقدار بازگشتی را در انتهای تعریف بدنه تابع قرار می دهیم .

مثال : Function Add (var a1 , var a2) {

→ C معادل یک تابع در پاسکال
= a1+a2;

return C; }

Function Show (var msg) { معادل پروسیجر در پاسکال

Document . write (msg); }

K = Add (نحوه استفاده از تابع Add در خارج از تابع .
A , B)

مثال :

```
<script> function show() {
```

```
    Document element by Id ("az").inter html= '<input
```

```
        type ="button" value ="send" name ="n" />'
```

```
    }
```

خروجی

| | |
|--|----------|
| | Click me |
|--|----------|

با کلیک روی دکمه پیغام send ظاهر می شود .

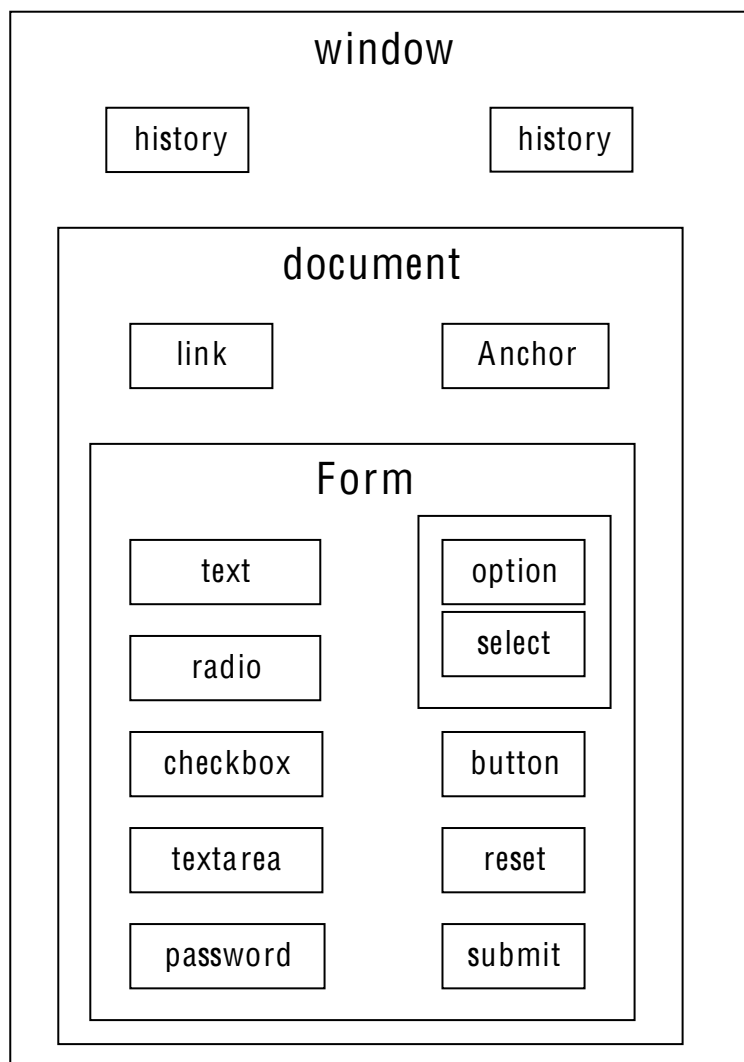
خصوصیت id :

برای تمامی عناصر داخل صفحه می توانیم خصوصیته به نام خصوصیت id تعریف کنیم به گونه ای که مقدار این خصوصیت یک عنصر را در داخل صفحه وب به صورت یکتا مشخص می کند از این خصوصیت برای ارجاع بهد یک عنصر در داخل کدهای javascript می توانیم استفاده کنیم .

DOM ۱۲

یک مدل شی گرا مستقل از یک بستر خاص و مستقل از یک زبان برنامه نویسی خاص براساسی عناصر داخل یک صفحه وب است.

مدل : نمایش خلاصه شده چیزی لست که جزئیات آن را حذف کردیم



با استفاده از DOM یک مفهوم سلسله مراتبی برای عناصر داخل صفحات وب تعریف می شود ریشه ساختار سلسله مراتبی در یک صفحه شیء windows هست و تمامی اشیاء بعدی زیر شاخه windows خواهند بود هر شیء در DOM معادل یا به نوعی معرف یک عنصر یا یک مفهوم مرتبط با صفحه web است .

به عنوان مثال :

Windows : بیانگر و معرف پنجره Browser وب است .

History : معادل مفهوم history در مرورگر وب است

Document : معادل صفحه وب

Link ,Anchor : معادل لینکهای صفحه وب (لنگرگاه = Anchor)

Form : معادل فرم های داخل صفحه وب است

نکته : این مدل یک مدل سلسله مراتبی است و در این جا window ریشه است و بقیه فرزندان آن است.

مثال : پنجره browser را می بندد
window . close () ;

```
< input onclick = window . close () value = " close " />
```

1) window.history

اشاره به لیست history مربوط به Browser.

اگر بعد از این back بنویسیم برمی گردد به صفحه قبل.

2) window.document

اشاره به متن سند جاری (همان صفحه html) که در داخل browser باز شده است.

3) window.location

به آدرس browser اشاره می کند.

```
< input onclick = " window.location " value=yahoo location="http://www.yahoo.com " />
```

یعنی اشاره به form یا form یا image یا window . document . link

تمامی عناصر داخل صفحه html زیر شاخه شيء document موجود در سلسله مراتب DOM هستند .

نام

window . document . form . text box

اگر می خواهیم به یک text box داخل صفحه داشته باشیم

مثال : دسترسی به یک text box در داخل یک فرم در صفحه html از طریق سلسله مراتب DOM

```
Window . document . form . text 1 . value = " Ali " ;
```

مقدار داخل جعبه متن جعبه متن با نام text1 فرم داخل وب صفحه html پنجره browser

اینها در سمت client انجام می گیرد قبل از ارسال به سمت سرور پس می توانیم در value شرط بگذاریم که به طور مثال اگر قسمتی از form خالی بود (در پر کردن توسط client) یک پیغام برایش بنویس یا submit نکن .

```
If value == " Ali " then ....
```

رویدادها^{۱۳} events

تعاملاتی که یک کاربر با صفحه html دارد می تواند منجر به ایجاد یک رویداد شود با استفاده از زبان جاوا اسکریپت می توانیم به این رویدادها پاسخ دهیم .

- رویدادهای تعریف شده برای تمامی عناصر در داخل صفحه یکسان نیستند .

مثال : رویداد onclick

وقتی روی می دهد که روی یک عنصر کلیک موس انجام می گیرد .

- رویدادهای عناصر داخل صفحه را می توانیم به کدهای java Script انتصاب دهیم. در این صورت زمان اتفاق افتادن آن رویداد کد java Script منتصب به آن رویداد اجرا خواهد شد.

مثال رویدادها:

onBlur : زمانی اتفاق می افتد که یک عنصر focus را از دست می دهد. مرکزیت را به دست می گیرد یعنی سو در مرحله قبل از اجرا در Bottom مرحله های لایت قبل.

```
<input onblur = " window.close() " value = "close ">
```

در اینجا اگر روی کلیک close کلیک کنیم اتفاقی نمی افتد بلکه اگر از آن خارج شویم close می کند.

<a> , <area> , <button> , <input> , <lable> , <select> , <textarea> ,
<apple> , <div>

OnChange : زمانی که مقدار یک عنصر تغییر می کند.

<input> , <select> , <textarea>

On focus : زمانی اتفاق می افتد که رویدادی focus می گیرد و می شود بر عکس onblur

<a> , <area> , <button> , <input>

Onload : زمانی که عنصر در صفحه load می شود.

On Mouse over : زمانی که اشاره گر ماوس بر روی عنصر وارد می شود.

<apple> , , <ilayer> , <layer>

On Mouse out : زمانی که اشاره گر ماوس از روی عنصر خارج می شود.

<apple> , , <ilayer> , <layer>

: On Mouse up

<apple> ,

: On Mouse move

<apple> ,

On key down : زدن کلید

On key press : زدن و رها کردن کلید

On submit : زمانی که فرم ارسال می شود.

<form>

On Reset : زمانی که اطلاعات عناصر داخل فرم Reset می شود.

On select : زمانی که یک عنصر از یک List انتخاب می شود.

<input> , <textarea>

On Unload : زمانی که یک عنصر از حالت load خارج گردد عکس on load

<body> , <frameset>

زبان CSS^{۱۴}:

زبان html برای مشخص کردن قالب ، فرمت و خصوصیات محتوای خود دارای ضعفهایی است برای این که بتوانیم خصوصیت‌های عناصر زبان html را به صورت دقیق مشخص کنیم می توانیم از زبان CSS استفاده کنیم.

به سه طریق می توان از CSS در یک صفحه html استفاده کرد :

1- با استفاده از خصوصیت style مربوط به تگهای html به صورت زیر :

```
< نام tag / > محتوای tag > " کد CSS " style =
```

نام tag

"... ، مقدار : نام خصوصیت ، مقدار : نام خصوصیت"

2- کدهای مربوط به CSS در داخل تگ style در همان صفحه قرار می گیرد و سپس برای ارجاع به آن کدها می توانیم از خصوصیت class مربوط به تگهای html استفاده کنیم .

```
< style >
```

```
. text 1 :{
```

```
Color : read ;
```

```
Font : Tahoma {
```

```
< /style >
```

```
< نام tag / > محتوای tag > " text 1 " class =
```

```
< نام tag
```

3- کد CSS را در یک فایل جداگانه قرار دهید و سپس از صفحه html یک ارجاع به آن فایل ایجاد نمایید و برای استفاده از آن CSS می توانیم همانند مورد قبل از خصوصیت class مربوط به تگهای html استفاده نماییم .

برنامه نویسی سوکت^{۱۵}

تعریف سوکت (پورت)

در یک شبکه ممکن است درخواستهای مختلفی به یک host برسد سیستم عامل host مذکور برای اینکه بتواند تشخیص دهد درخواستهای رسیده باید توسط کدام نرم افزار هندل شود (کنترل و مدیریت شود) از پورتها استفاده می کند به عبارت دیگر پورتها در یک کامپیوتر آدرس دهی دقیق نرم افزاری که به آن (assign)^{۱۶} شده است را نشان می دهد.

نسبت دادن یک نرم افزار به یک port

وقتی نرم افزار یا سرویسی را به یک port در یک کامپیوتر نسبت می دهیم به سیستم عامل می گوییم که درخواستهایی را که برای آن پورت خاص از طریق شبکه وارد می شود را به نرم افزار مورد نظر ما ارجاع دهد و یا آن را مطلع نماید.

ساختار یک برنامه server در برنامه نویسی سوکت :

1- یک object از نوع server socket برای یک پورت خاص تعریف می شود.

```
listen socket = new Server socket جاوا
```

```
Server socket (server port)
```

```
Dim listen socket as server socket = new
```

```
Server socket (server port)
```

2- ایجاد یک ساختار تکرار که همواره در حال اجرا بوده و درخواستهای رسیده را دریافت کند.

15 - socket programming

نسبت دادن - 16

```
While (true) {
```

```
Socket client socket = listen socket. accept ();
```

3 { يك انتظار براي دريافت يك درخواست
عمليات بعدي با
listen socket در پورت مشخص شده در
استفاده از

```
client socket ( 7777 پورت )
```

```
}
```

3- قسمت مربوط به برنامه توليد پاسخ براي درخواست ارسال شده :

اين قسمت بهتر است كه به صورت **thread** (نخ) نوشته شود
اگر اين قسمت به صورت **thread** نوشته نشود درخواستهاي
رسيده به سرور نمي تواند به صورت موازي مورد
پردازش قرار گرفته و پاسخ مناسب براي آنها توليد
شود لذا قابليت **thread** اين امكان را مي دهد كه ما
براي درخواستهاي رسيده پاسخهايي را به صورت موازي
و همزمان توليد كنيم .

نكته : **thread** : به زبان ساده **thread** ها كدهايي هستند
كه مي توانند به صورت موازي در داخل يك پروسه
(برنامه) اجرا شوند .

```
Receiver r = new receiver ( client socket ) r. run ( );
```

Thread را اجرا مي کند

يك كلاس از نوع **thread** است .

```
Server socket listen socket = new server socket ( server port )
```

7777

```
While ( true )
```

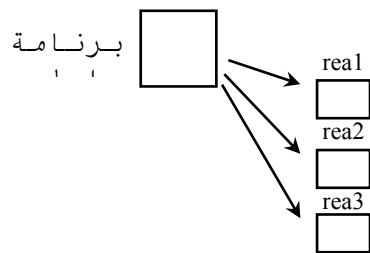
```
Socket client socket = listen socket. accept ( );
```

```
Receiver r = new receiver ( client socket ) r. run ( );
```

نکته : وقتی اجرای یک thread شروع می شود ادامه

اجرای آن thread می تواند مستقل از برنامه اصلی

ایجاد کننده آن thread ادامه یابد.



نخها یا thread

وقتی run اجرا می شود این run 20 شاخه می شود یکی

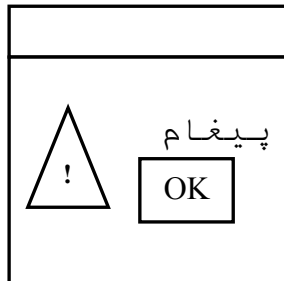
یک thread ایجاد می کند و دیگری منتظر درخواست

بعدي می شود.

پنجره های ازپیش تعریف شده در java Script

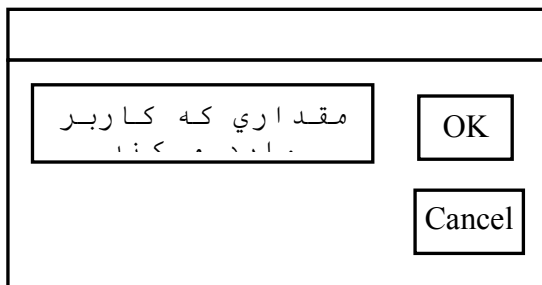
Alert -1

یک پیغام را برای کاربر نمایش می دهد.



prompt _ 2

می توان دو مقدار گرفت. با استفاده از یک پنجره یک مقدار را از کاربر می گیرد.

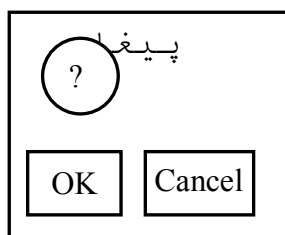


بالای text book می آورد. $C = \text{prompt}(\text{"Enter your name"})$

- مقدار بازگشتی این تابع مقداری است که توسط کاربر داخل پنجره وارد می شود.

confirm _ 3

برای گرفتن تأیید از کاربر



- مقدار بازگشتی آن True یا False است.

$K = \text{confirm}(\text{"پیغام"})$

Ok → True

Cancel → False

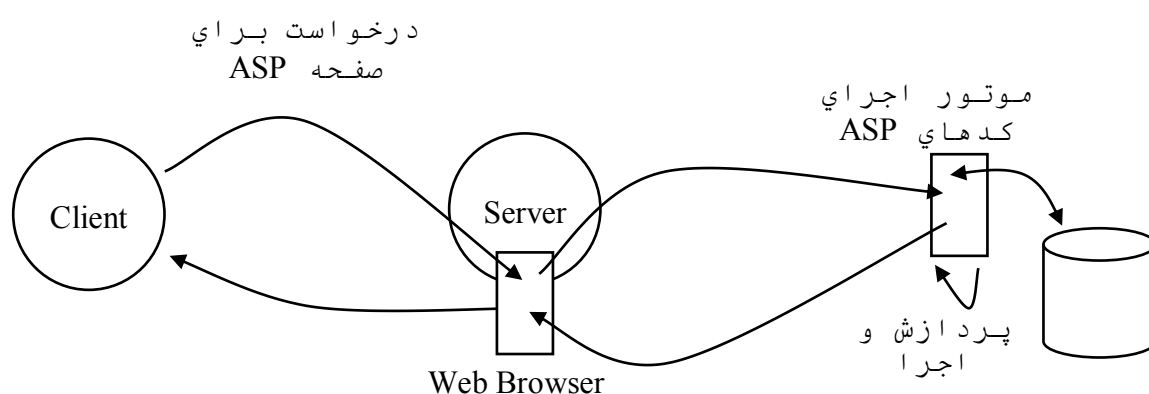
برنامه نویسی سمت سرور

تکنولوژی ASP^{۱۷}:

تکنولوژی توسط مایکروسافت برای توسعه برنامه های تحت وب معرفی شده است.

صفحاتی که دارای کد ASP هستند زمانی که از سمت client درخواست می شوند ابتدا کدهای ASP اجرا شده سپس نتیجه به client ارسال می گردد.

• صفحاتی که دارای کد ASP هستند به جای پسوند htm یا html دارای پسوند asp هستند.



خواندن از روی دیسک

• کدهای نوشته شده در asp با استفاده از زبان vb Script است.

زبان vb.Script شبیه زبان ویژوال بیسیک است. این زبان شیء گران بوده ولی خود از اشیاء از قبل تعریف شده استفاده می کند.

vb.Script یک زبان کامل نیست بلکه یک زبان اسکریپتی است برای برطرف کردن نقص موجود در زبانهای اسکریپتی برای انجام بعضی از کارها در asp از قطعاتی که در سایر زبانها مانند ویژوال بیسیک ، C++ و ... نوشته شده می توانیم به صورت اشیاء COM در asp استفاده کنیم.

نکته: asp یک زبان نیست بلکه یک تکنولوژی ایجاد صفحات پویای سمت سرور است.

طریقه نوشتن کدهای asp در داخل یک صفحه

1 – در بین 2 تگ script به صورت زیر است:

```
<% = <script language = " VB.Script " runat = "server ">
```

کدهای VB.Script (کدهای asp)

```
%>=</script>
```

2 – بین دو علامت <% و %> به صورت زیر

```
<% asp کدهای %>
```

```
<%
```

```
Response,write("this code is generated by asp")
```

```
%>
```

3 – اگر یک کد یک خطی داشته باشیم کد خروجی آن رشته باشد می توانیم به صورت زیر هم آن را بنویسیم

کد %>

<%= asp

: < input type = " text " value + "<% = email %> "/>

مثال

متغییری که قبلاً در داخل صفحه تعریف شده است.

این کدها را سرور اجرا می کند و بعد حذف می کند و

کاربر آنها را نمی بیند یعنی کد تبدیل می شود به :

<input type = " text " value = "ali @ yahoo.com"/>

یعنی عبارت بالایی جایگزین می شد با این عبارت توسط

سرور

کاربر نمی بیند که صفحه چیست یعنی نمی داند asp است

یا php یا ... بلکه فقط نهایی را می بیند یعنی html

را می بیند. مفهوم پردازش هم همین است.

اشیاء پر کاربرد و اصلی مورد استفاده در asp :

زبان asp یک زبان شی گرا نیست اما اشیای در داخل

این زبان وجود دارد که می توانیم از آنها استفاده

کنیم این اشیا عبارتند از :

Request – 1 درخواست

Respons – 2 پاسخ

Application – 3 برنامه

Session – 4 جلسه

Server – 5 خادم (سرویس دهنده)

: Request

این object نماینده درخواست ارسال شده به سمت سرور می

باشد بنابراین با استفاده از آن می توانیم به

اطلاعاتی نظیر داده های وارد شده در داخل فرم های وب ، نوع مرور گری که این درخواست از آن میسر و... دست پیدا کنیم .

به عنوان مثال : مقادیر فرمهایی که از سمت client به سرور ارسال می شود از طریق این object قابل دسترسی است .

: Respons

این object نماینده پاسخ ارسال شده به سمت client است

توسط این شیء می توانیم پاسخی را که به client ارسال می کنیم تغییراتی در آن ایجاد کنیم و به توابع مربوط با سروری که برنامه asp بر روی آن اجرا می شود دسترسی داشته باشیم .

به عنوان مثال می توان برای نوشتن کوکی ها در سمت client ف هدایت client به یک صفحه دیگر ، نوشتن سر آیند های پروتکل HTTP و... استفاده کرد .

مثال :

```
<%
```

```
Response.redirect "http://www.w3schools.com"
```

```
%>
```

این کد باعث می شود کاربر به طور اتوماتیک به سایت <http://www.w3schools.com> هدایت شود .

```
<%
```

```
Response.Cookies ("firstname")="alex"
```

```
Response.Cookies ("firstname").expires=# may 10.2008 #
```

%>

این کد باعث می شود کوکی با نام `firstname` و محتوای `alex` ایجاد شود و `may 10.2008` زمان از دست رفتن کوکی را مشخص می کند

3 Application : این `object` نشان گر برنامه ای است که بر روی سرور اجرا می شود.

نکته : متغیرهایی که در داخل این `object` تعریف شوند توسط تمامی صفحات `asp` موجود بر روی سرور قابل دسترسی خواهند بود. به عبارت دیگر متغیرهای تعریف شده در این `object` ، سراسری (`global`) هستند.

مثال :

```
<script runat = "server" language = "vb script">
```

```
application ("count2") = application ("count2")+1
```

```
b = Session ("count")
```

```
</script>
```

```
Global visit count is <% = b %>
```

```
Application (" name ") Ali
```

در صفحه دیگر می نویسیم

Session – 4

نماینده جلسه کاری است

• مفهوم session یا جلسه

پروتکل `http` هیچ سابقه ای در مورد درخواستهایی که از سمت یک `client` به سرور ارسال می شود نگه نمی دارد. برای این کد در برنامه های تحت وب نیاز داریم تا سابقه درخواستهای آمده از یک `client` خاص را در برخی

موارد نگهداري كنيم. مفهومي به نام session مطرح مي شود. مفهوم session توسط تكنولوژيهايي كه براي برنامه نويسي سمت سرور مورد استفاده قرار مي گيرند پياده سازي مي شود.

در يك برنامه تحت وب زماني كه در خواستي از يك کاربر به سمت سرور ارسال مي شود اصطلاحاً يك جلسه كاري براي آن كار بر ايجاد مي گردد اين جلسه كاري مختص آن کاربر است يا به عبارتي هر کاربر جلسه كاري مخصوص به خود را دارد به عنوان مثال اگر متغيري در داخل يك Session قرار بگيرد مقدار آن متغير الزاماً براي تمامي كاربران يكسان نخواهد بود. اگر تا يك مدت زمان مشخص (قابل تعريف توسط برنامه نويس) درخواستي از آن client به سرور ارسال نگردد آن session از بين خواهد رفت.

• مفهوم Session Time out

در واقع مدت زماني است كه اگر پس از طي اين مدت زمان درخواستي از client ارسال نشود session مربوط به آن client از بين خواهد رفت. با توجه به توضيحات اخير در يك برنامه سمت سرور به ازاي هر client يك session وجود خواهد داشت.

Object Session : اين شيء بيانگر session ي ايجاد شده در برنامه سمت سرور است. اگر متغيري را در يك شيء session ذخيره نماييم تنها در محدوده آن session قابل دسترس خواهد بود.

مثال :

```
<script runat = "server" language = "vb script">
```

```
Session ("count") = 3
```

```
Session ("count") = Session ("count") + 1
```

```
a = Session ("count")
```

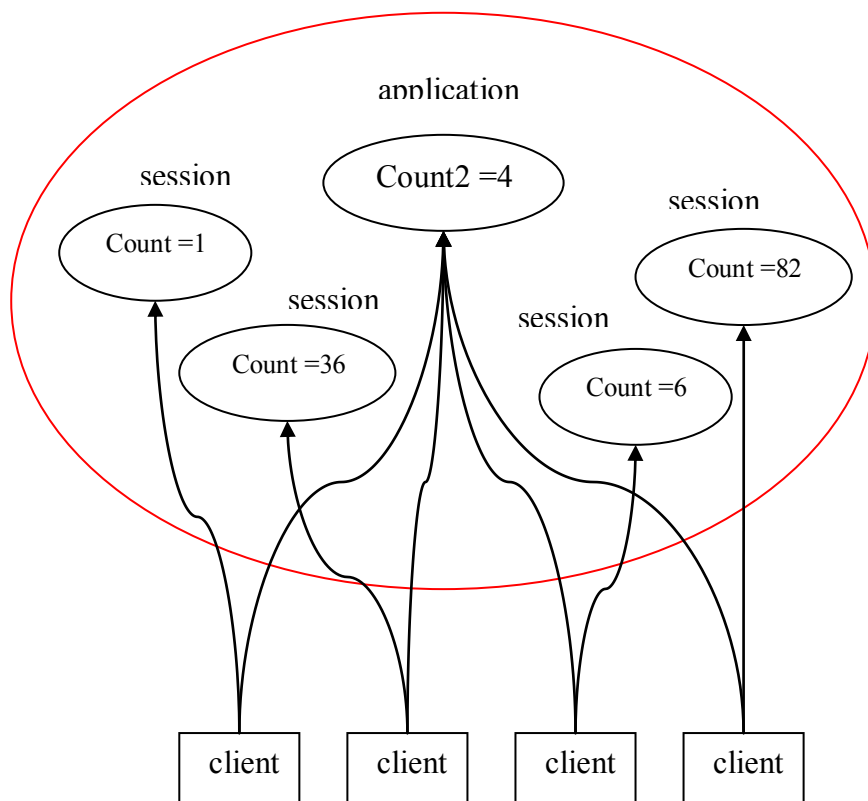
```
</script>
```

```
Your visit count is <% Session ("count") %>
```

```
<% response . write (" رسته ") %>
```

نکته : به ازاء هر client يك session هست ولي به ازاء تمام client ها يك Application است . مثلاً اگر در yahoo ما log out مي كنيم مي گوييم : session را حذف كن و يا اگر بعد از آخرين ارتباط در زمان log in مدت زمان time out سپري شود خودبخود session را حذف مي كند و اگر بخواهد استفاده كنيد از شما log in مي خواهد .

برنامه سمت سرور



• مفهوم Cookie ها (کوکي) :

کوکيها اطلاعات متني هستند که معمولاً توسط سرور توليد شده و به client ارسال مي گردد. Client ها (browser) به صورت اتوماتيک اين اطلاعات را گرفته و آنها را ذخيره مي کند و سپس درخواستهاي بعدي که به همان سرور ارسال کننده کوکي ارسال مي شود کوکيهاي ذخيره شده را همراه درخواست جديد به سرور ارسال مي کند.

نکته : یکی از راهکارهای پیاده سازی مفهوم session توسط تکنولوژی‌های برنامه سازی سمت سرور استفاده از کوکی می باشد.

نکته : برای هر کوکی می توان زمانی را که client باید آن را نگهداری کند را تنظیم نمود.
اگر بخواهید کوکی را پاک کنید

IE → Tools → Internet → option → General

آن موقع دوباره از شما log in می خواهد. البته اگر تکنولوژی برقراری session توسط کوکی انجام گرفته باشد.

نکته : در asp.net توسط object ، request می توان به کوکی‌های ارسال شده از سمت client دسترسی داشت و توسط object response می توان کوکی‌هایی را که به client ارسال می شود کنترل نمود.

دسترسی داریم به کوکی‌ها
request . cookies

دسترسی داریم به کوکی‌های ارسال شده .
response .
cookies

نکته : مربوط به
object request

برای دسترسی به اطلاعات وارد شده در داخل فرم‌های وب می توانیم از مجموعه‌های Form و Guary string تعریف شده در داخل object request استفاده کنیم.

Method: get → query string post → form

< Form Method = post / get >

! اگر method = post مجموعه اطلاعات ارسالي را با form مي‌گيريم .

! اگر method = get مجموعه اطلاعات ارسالي را با query string مي‌گيريم .

5 _ Server

نماينده سروري است که برنامه تحت وب بر روي آن اجرا مي‌شود به عنوان مثال از مدت transfer مربوط به اين object مي‌توانيم براي هدايت پردازش صفحه asp به يك صفحه ديگر استفاده كنيم .

مثال:

```
User = request .from ("user")
```

```
pass = request .from ("pass")
```

```
if (pass = "123456") then
```

```
server . transfer ("a9.asp") -----> <% = session ("name")%>
```

```
else
```

```
server . transfer ("a10.asp")
```

```
end if
```

: ADO

براي دسترسي به پايگاه داده در ASP از تکنولوژي ADO استفاده مي‌کنند. در واقع با استفاده از ADO مي‌توانيم يك اتصال به پايگاه داده برقرار نموده داده‌ها را در داخل آن ذخيره نماييم و در مواقع لزوم اين داده‌ها را باز يابي كنيم. براي استفاده از ADO در ASP ابتدا با يدي اتصال (که يك Object

تعریف شده در ASP است) ایجاد نمایم. سپس یک دستور (command ...) (یک Object تعریف شده در ASP) ایجاد نموده و دستورات SQL را در داخل Object ، command قرار می دهیم و سپس این Object ، command را از طریق اتصال تعریف شد ه قبلی به پایگاه داده ارسال می کنیم. با استفاده از connect ، command را به پایگاه داده ارسال کرده. ! با استفاده از Object ، parameter می توانیم پارامترهایی را به دستورات ارسالی به پایگاه داده تعریف نمایم.

مثال: در برنامه ای که نام یک شخص را می گیرد و اطلاعات آن شخص را از پایگاه داده بازیابی می کند در دستور SQL به کار رفته در Object ، command مربوط به این بازیابی اطلاعات می توانیم نام شخص را به عنوان پارامتر تعریف نمایم. تا در هر باری که یک نام توسط کاربر وارد می شود این نام را به عنوان پارامتر به پایگاه داده برای اجرای دستور تعریف شده ارسال نمایم.

: Record set

زمانی که دستوری را به پایگاه داده ارسال کردیم نتایج باز گردانده شده از پایگاه داده در قالب یک Object به نام Record set در ADO قرار می گیرد.

: ملزومات ایجاد یک برنامه ASP :

1. یک web server که از تکنولوژی ASP پشتیبانی می کند. این web server معمولاً « IIS^{۱۸} » می باشد که در

سیستم عامل‌های ویندوز 2000 به بعد می‌تواند نصب شود. در ویندوز 98 برای اجرای برنامه‌های ASP باید PWS^{۱۹} را نصب کرد.

2. یک ویرایشگر متن ساده مثل Not pad ویرایشگرهای پیشرفته‌تری که کار نوشتن برنامه‌های ASP را راحت‌تر می‌کنند نیز در بازار وجود دارند. مثل Dreamweaver.

! برنامه‌های نوشته شده در ASP را با پسوند .asp ذخیره می‌نماییم. این فایل‌ها برای اینکه بتوانند اجرا شوند باید داخل پوشه‌ای که توسط « IIS » web server مشخص می‌شود قرار گیرند.

تکنولوژی (PHP) Personal Home Page :

PHP یک زبان اسکریپتی سمت سرور برای ایجاد برنامه‌های تحت وب است. PHP یک زبان opensource است و کدهای آن در دسترس است نسخه جاری آن 5 بوده که از شی‌گرایی به طور کامل پشتیبانی می‌کند.

صفحات PHP را هم می‌توان بر روی سیستم عامل ویندوز و هم بر روی سیستم عامل‌های مبتنی بر Unix مانند linux اجرا کرد.

نحو (syntax) زبان PHP :

نحو (syntax) زبان PHP بسیار شبیه زبان C می‌باشد

نحوه قرار دادن کدهای PHP در داخل یک صفحه :

پسوند صفحات حاوی کدهای PHP :

PHP3 . یا PHP . یا PHtml .

19. Microsoft personal web server

```
<?php
```

```
<?
```

یا ← حالت

کدهاي php

```
php
```

استاندارد

```
?>
```

```
?>
```

مثال :

```
<?
```

```
//this is php demo
```

```
Echo "welcome to php world"
```

```
?>
```

متغیرها در زبان PHP :

زبان PHP در مورد نوع متغیرهای تعریف شده زیاد سخت گیر نیست علاوه بر آن متغیرها را بدون آنکه به صورت صریح استفاده کنیم می توانیم مورد استفاده قرار دهیم .

متغیرها در زبان PHP با یک علامت \$ شروع می شوند .

مثال :

```
<? Php
```

```
$ name = "ali " ;
```

```
$score = "18"
```

```
?>
```

```
<?php
```

```
$fname="hassani"
```

```
Echo $name.$fname . "is student";
```

?>

خروجي : Ali hassani is student

عملگر ها در PHP :

- عملگر هاي رياضي : + ، - ، ++ ، -- ، / ، * ، %
- عملگرهاي مقايسه اي : != ، < ، > ، <= ، >=
- عملگر هاي منطقي : ! (not) ، && (and) ، || (or)

ساختار هاي شرطي و تکرار :

ساختار شرطي if/else :

```
if ( شرط )
{
    (بدنه شرط)
}
Else
{
    ( بدنه else )
}
```

مثال :

```
if (شرط)
Echo "welcome ali" ;
}
Else { echo "incorrect password " ;}
```

ساختار تکرار : for

```
For ($i=1 ; $i <= 10 ; $i++)  
{  
( بدنه for )  
}
```

مثال :

```
Echo "<h" . $si ."> welcome to php ! </h" . $i . "> <br>" ;
```

نحوه دسترسی به مقادیر فرم های وب :

- **\$_GET** : برای فرم هایی که متد مربوط به فرم get است

- **\$_POST** : برای فرم هایی که متد مربوط به فرم post است

- **\$_REQUEST** : هم برای متد get و هم برای متد post است

مثال :

```
$name = $_GET [ "n" ]
```

```
$name = $_REQUEST [ "n" ]
```

می خواهیم اطلاعات را از فرم بگیریم :

```
$name = $_post [ "name" ]
```

```
$fname = $_post [ "fname" ]
```

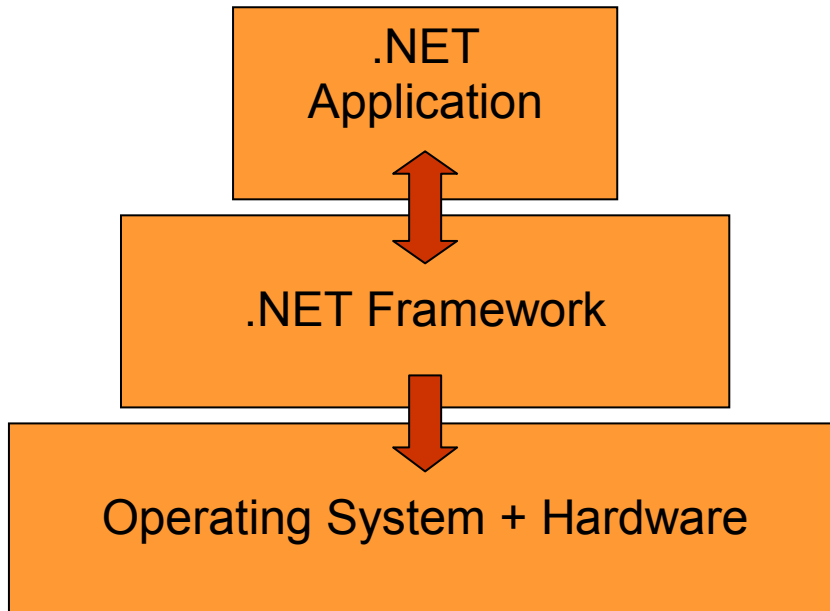
```
$tel = $_post [ "tel" ]
```

```
Echo "name" . $name . "<br/>" ;
```

```
Echo "fname" . $fname . "<br/>" ;
```

```
Echo "tel" . $tel . "<br/>" ;
```

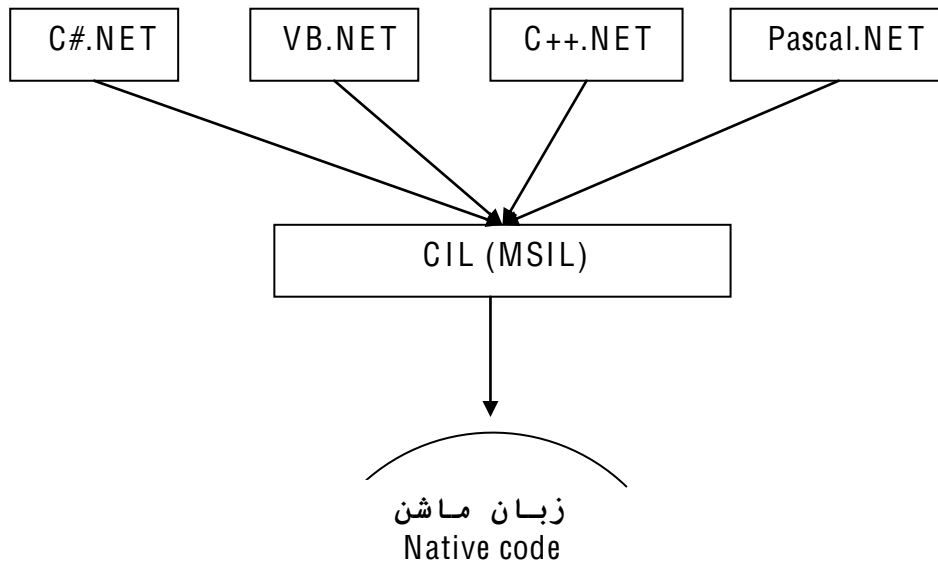
تکنولوژی .NET :



در سال 2001 این تکنولوژی توسط شرکت مایکروسافت ارائه شد.

.net یک سکوی نرم افزاری برای تکنولوژی یک بستر یک پارچه برای توسعه برنامه‌های تحت وب و همچنین سیستم عامل های windows مایکروسافت جهت توسعه برنامه های کاربردی در سطح یک سازمان است ..

چهار چوب .NET :



CL :۲۰

یک زبان مشترک سطح پایین است که تمامی کدهای کامپایل شده در بستر .net به جای زبان ماشین به این زبان تبدیل می شوند . بنابراین کدهای نوشته شده در بستر .net مستقیماً به زبان ماشین تبدیل نمی شوند .

مزیت وجود (MSIL) CL :

وجود cl و یک تعریف استاندارد برای آن این امکان را به توسعه دهندگان نرم افزار می دهد که برای توسعه نرم افزار از زبان های مختلف برنامه نویسی که در بستر .net تعریف شده اند برای توسعه یک پروژه واحد استفاده نماییم . بدین معنی که به راحتی می

توان از کدهای نوشته شده در یک زبان خاص در یک زبان دیگر استفاده نمود.

نکته :

وقتی برنامه ای در .Net کامپایل می شود و فایل dll آن در ویندوز تولید می شود این فایل در واقع کدهایی در زبان cl است برخلاف فایل های dll سنتی که حاوی دستورالعمل های زبان ماشین بودند. زبان هایی که در .Net پشتیبانی می شوند عبارتند از :

VB . Net , VC++ .Net , C#.Net , pascal , ...

که به تدریج زبان های دیگری نیز به این مجموعه اضافه می شوند .

Base Class Library

کتابخانه اصلی بستر .net (BCL)^{۲۱}

چهار جوب NET. مجموعه کدهای کتابخانه ای آماده که کارکرد های مفید برنامه نویسی را پیاده سازی کرده اند تحت عنوان (BCL) جهت تسریع کار برنامه نویسان فراهم کرده است .

با توجه به ساختار NET. تمام زبان های بستر NET. می توانند از BCL استفاده کنند .

کارکرد هایی مثل خواندن و نوشتن از ورودی و به خروجی ف واسط های برنامه نویسی برای عملیات

21. Base class library

گرافیکی ، ارتباط با پایگاه داده ، کار با XML و
.... را دارد

نکته : زبان C# به عنوان زبانهای پیشنهادی
مایکروسافت و زبان اصلی Net. شناخته می شود.

مزایای بستر Net :

1. از زبانهای برنامه نویسی مختلفی پشتیبانی می
کند لذا از قابلیت‌های زبانهای مختلف برنامه نویسی
برای توسعه برنامه‌ها می توانند استفاده کنند همچنین
در صورت مدیریت صحیح افراد سیستم توسعه را می
توان با مهارت‌های برنامه نویسی مختلفی مورد
استفاده قرار داد .

2. بستر Net. مجموعه کامل و قدرتمندی را تحت عنوان Base
Class Library در اختیار بستر Net. نویسان برای نوشتن
برنامه‌های خود قرار می دهد .

3. محیط توسعه یکپارچه ای مانند VS .Net که تقریباً
تمام مراحل توسعه یک برنامه ی کاربردی را پشتیبانی
می کند برای آن وجود دارد.

4. یک راه حل قوی و آسان به نام تکنولوژی asp.Net
برای توسعه برنامه‌های تحت وب ارائه می کند .

معایب تکنولوژی Net :

1. به صورت انحصاری در اختیار شرکت مایکروسافت است
و کدهای اصلی بستر Net. در اختیار همه قرار ندارد .
(open source نیست)

2. برنامه‌های نوشته شده در بستر Net. تنها قابلیت اجرا در بسترهای تولیدی شرکت مایکروسافت را دارد به عبارتی دیگر بستر Net. تنها می‌تواند بر روی سیستم عامل ویندوز نصب شود لذا برنامه‌های نوشته شده نیز الزاماً تنها بر روی این سیستم عامل قابلیت اجرا دارند.

پیاده سازی بستر NET. غیر از Microsoft.NET :

(1) NET. porto Gle : که جزوی از پروژه GNU. است این پروژه هدفش تولید یک نرم افزار opensource برای اجرای برنامه های کامپایل شده به زبان CIL در بستر های نرم افزاری مختلف از جمله linux می باشد .

(2) Mono project :

این پروژه توسط شرکت novel هدایت می شود و هدف آن پیاده سازی محیط اجرای برنامه های NET. و همچنین کتابخانه استاندارد چهار چوب NET. می باشد. در حال حاضر پشتیبانی از ado.NET , asp.NET در این پروژه فراهم شده است و کامپایلر های زبان C#.NET , VB.NET پیاده سازی شده است.

تکنولوژی ASP.Net

تکنولوژی‌ای که در بستر Net. برای توسعه برنامه‌های تحت وب توسط مایکروسافت معرفی شد. این تکنولوژی نه یک ورژن جدید از تکنولوژی asp بلکه معرفی یک مفهوم و ساختار جدیدی برای نوشتن برنامه‌های تحت وب بود. در ASP.Net برخلاف تکنولوژی ASP از تمامی زبان هایی که بستر Net. آن را پشتیبانی می کند برای نوشتن برنامه های ASP.Net می توانیم استفاده کنیم .

زبانهاي بستر .NET. زبان هاي كاملي هستند يعني قابليت هاي گسترده تري نسبت به زبان هاي اسكريپتي مانند jscript , VB script که براي برنامه هاي ASP استفاده مي شد ارائه مي دهند

خصوصيات و مزاياي تکنولوژي asp.Net نسبت به asp :

1. استفاده از يك زبان برنامه نويسي كامل : برخلاف asp که از يك زبان اسكريپتي که براي نوشتن کدهاي آن استفاده مي شود در asp.Net از يك زبان برنامه نويسي كامل براي نوشتن برنامه هاي تحت وب استفاده ميکنيم .
2. پشتيباني از زبانهاي برنامه نويسي مختلف : به سبب اين که asp.Net بر پايه تکنولوژي .Net (.Net Framework) بنا نهاده شده است لذا همانند ساير برنامه ها در asp.Net مي توان از زبانهاي برنامه نويسي مختلف استفاده نماييم .
3. يك محيط توسعه يك پارچه و قوي و ويژوال براي توليد برنامه هاي asp.Net به نام VS .Net وجود دارد .
4. وجود يك سري ابزارهاي آماده براي نوشتن برنامه هاي asp.Net

مثال : ابزارهايي مانند File uploade , ...

,Data Grid

5. asp.Net اين امکان را فراهم مي آورد که کدهاي برنامه نويسي را از طراحي صفحه جدا کنيم در حالي که در asp کدهاي برنامه نويسي و کدهاي صفحه در داخل همديگر استفاده مي شوند اين امکان کمکي به تقسيم کار در پروژه هاي تيمي است .

6 . امکان توسعه قطعات نرم افزاري توسط شرکت هاي مختلف جهت استفاده در برنامه هاي asp.Net به صورت استاندارد فراهم شده است .

ADO.Net

همانند تکنولوژی asp.Net این تکنولوژی نیز نوعی رویکرد جدید برای کار با پایگاه داده در بستر .Net ارائه می کند . VS .Net^{۲۲}

یک محیط توسعه یکپارچه IDE^{۲۳} برای توسعه برنامه های نوشته شده در بستر .Net ارائه می کند . VS.Net زبانهای VB , VC++ , VC# ,... را پشتیبانی کرده می توانیم کدها را در داخل این محیط ایجاد نمایم آنها را کامپایل کرده و سپس از طریق همین محیط اجرا کنیم در محیط VS.Net می توانیم صفحات asp (asp.Net) را نیز ایجاد کنیم . همچنین محیط VS .Net ابزارهایی برای کار کردن آسان با ADO.Net ارائه میکند . همانند سایر محیطهای توسعه امکان Debug کردن برنامه های نوشته شده حتی برای برنامه های ASP در محیط VS .Net وجود دارد .

Debug کردن : اشکال زدایی برنامه با بررسی خط به خط اجرای کدهای برنامه

! اجرای خط به خط کدها را trace هم می گویند .

نکته :

22. Visual Studio .Net

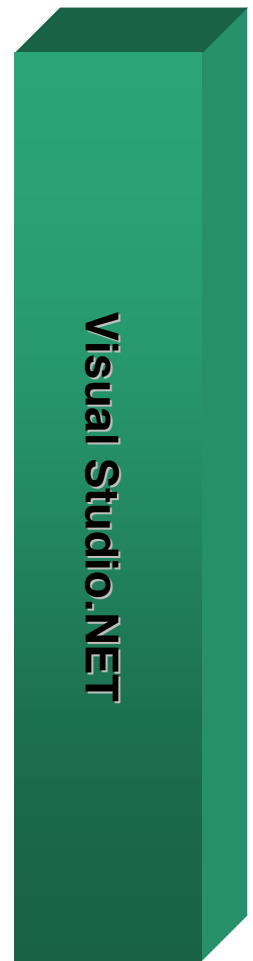
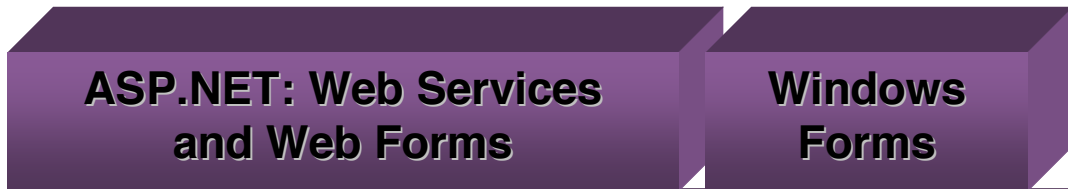
23. Integrated Development Environment

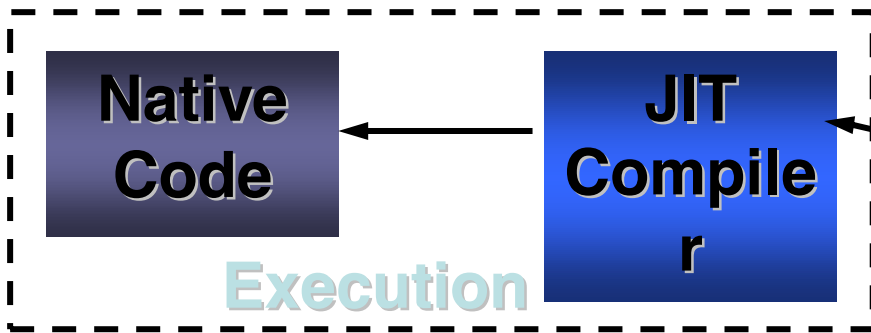
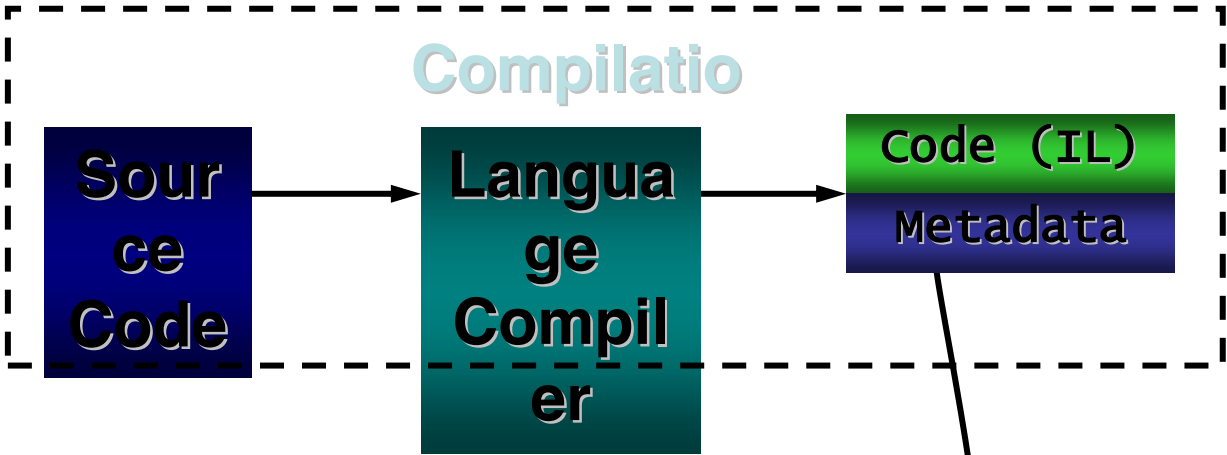
تمامی زبان‌های پشتیبانی شده در بستر .Net. زبان‌های
Object Oriented کلاسهای اصلی بستر .Net. با استفاده از
زبان C# نوشته می‌شود .

نکته :

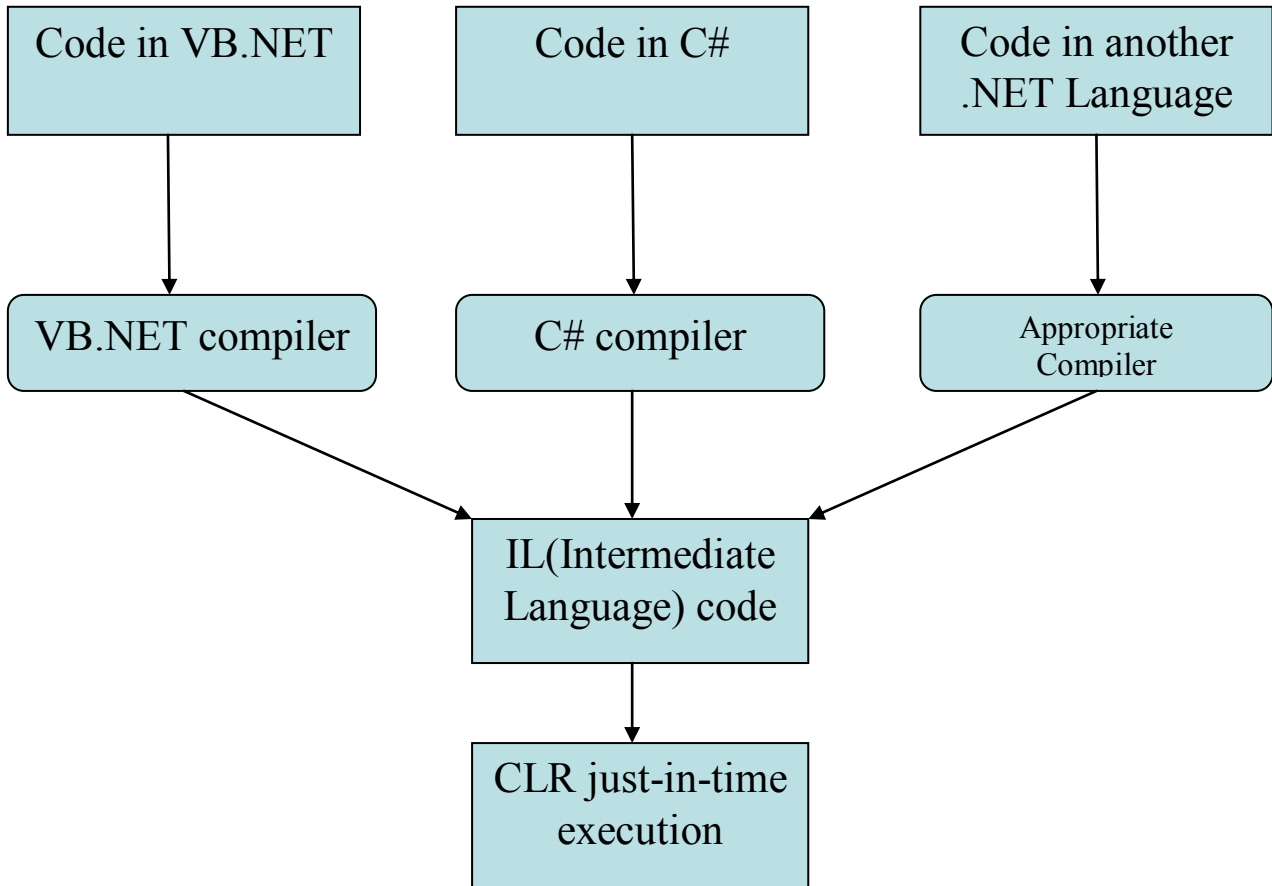
مزایای ADO.Net نسبت به ADO :

1. پایه و مبنای ADO.Net زبان نشانه گذاری XML
است. زبان XML یک توصیف استاندارد پر کاربرد برای
داده‌ها در محیط شبکه است.
2. روش اتصال disconnected را در پایگاه داده را
ارائه می‌دهد. در این متد برای دسترسی به داده‌های
موجود در پایگاه داده مورد نیاز نیست داده‌ها در
یک اتصال به داخل حافظه Load روش شوند
و درخواستهای بعدی از این داده‌های موجود در حافظه
می‌توانیم استفاده کنیم.
3. کلاسهای تعریف شده در ADO.Net ساختار بهتری نسبت
به کلاسهای موجود در ADO دارند.
4. به صورت کامل با بستر .Net سازگار است





At installation or the first time each method is called



وب سرویس‌ها :

در یک تعریف ساده وب سرویس‌ها برنامه‌های هستند که بر روی شبکه قرار می‌گیرند و سایر Host ها می‌توانند از خدمات ارائه شده توسط این برنامه‌ها استفاده نمایند. Host هایی که از این خدمات استفاده می‌کنند خود می‌توانند یک سرویس دهنده (سرور) باشند. که با استفاده از سرویس‌هایی برنامه مذکور سرویس‌های دیگری برای کاربران خود فراهم می‌نماید.

مفاهیم مرتبط با وب سرویس‌ها :

1. WSDL^{۲۴}

یک زبان مبتنی بر XML است که سرویس ارائه شده به صورت وب سرویس را توصیف می‌کند. به عنوان مثال : چه پارامترهایی از یک وب سرویس باید به آن ارسال کنیم ؟ پاسخ بازگشتی از وب سرویس به چه صورتی باشد ؟ و همچنین یک توصیف کلی از عملکرد توسط WSDL ارائه می‌شود.

2. SOAP^{۲۵}

این پروتکل یک پروتکل ارتباطی مبنی بر XML است که بای تبادل اطلاعات در بستر شبکه مورد استفاده قرار می‌گیرد ارتباطات بین Server , Client میزبان وب سرویس از طریق این پروتکل انجام می‌گیرد.

24. web service description language

25. Simple object access protocol

3. UDDI ۲۶

یک مکان ذخیره سازی خصوصیات وب سرویس‌های تولید شده در شبکه می باشد به عبارت دیگر اگر وب سرویس در شبکه تولید می شود برای آنکه پیدا کردن و استفاده از آن وب سرویس برای سایر Host ها میسر گردد معمولاً این وب سرویس‌ها را در محیطی به نام UDDI جمع آوری می کنند .

J2EE ۲۷

تکنولوژی J2EE یک بستر مناسبی است برای توسعه برنامه‌های سازمانی که به صورت توزیع شده در یک سازمان می تواند وجود داشته باشد . این بستر مشترک به ما این امکان را می دهد که قطعات مختلف نرم افزاری مورد نیاز در سازمان را با تکیه بر این بستر به صورت یکپارچه و سازگار با بقیه قطعات ایجاد شده مبتنی بر J2EE ایجاد نماییم .

! J2EE یک زبان برنامه نویسی نیست بلکه توصیف استاندارد است در مورد شیوه نوشتن برنامه‌های سازمانی به همراه کدهای برنامه نویسی و همچنین قطعات نرم افزاری که بر مبنای این توصیفات استاندارد ایجاد شده اند . این تکنولوژیک توسط شرکت سان میکرو سیستم معرفی شده اند (Sun microsystem) و تعداد زیادی از تولید کنندگان نرم افزار قطعات مختلف نرم افزاری خود را بر اساس این استاندارد ایجاد و عرضه می کنند .

زبان برنامه نویسی معرفی شده برای استاندارد J2EE
زبان java می باشد .

نکته : اخیراً شرکت Sun نام این تکنولوژی را از J2EE
به java EE تغییر داده است .

تکنولوژی‌های مرتبط با java2 :

سه بستر متفاوت در ارتباط با تکنولوژی java2 ()
نسخه دوم java (توسط شرکت Sun در سال 1999 معرفی
شده است :

1. J2SE^{۲۸}

برای توسعه برنامه‌های desktop (منظور برنامه‌هایی
که بر روی یک کامپیوتر اجرا می شوند و به صورت
توزیع شده و مرتبط با سایر قطعات نرم افزاری در
سطح یک سازمان نیست.

2. J2ME^{۲۹}

این نسخه برای ایجاد برنامه‌های java بطوری که
بتوانند بر روی ابزارهای کوچک غیر از کامپیوترهای
معمولی مانند موبایلها ، PDAS ، کارت‌های هوشمند
اجرا شوند ایجاد شده است .

3. J2EE^{۳۰}

توضیح قبلاً

برنامه‌های سازمانی :

برنامه‌های کاربردی سازمانی ، نرم افزارهایی هستند
که به صورت توزیع شده در سطح سازمان بر روی

28. Java 2 Standard Edition

29. Java 2 Micro Edition

30. Java 2 Enterprise Edition

Client های مختلف وجود دارند و انجام فعالیت‌های جاری در آن سازمان را تسهیل می‌کنند .

تکنولوژی‌های مرتبط با J2EE :

برخی از تکنولوژی‌هایی که در ارتباط با J2EE وجود دارند به صورت زیر معرفی می‌کنیم :

1. Servlets and JSP ^{۳۱}

این تکنولوژی در ارتباط با تولید صفحات پویای وب درست سرور معرفی شده است.

2. EJB ^{۳۲}

ساختار استاندارد برای کلاسها است که در بر گیرنده منطق و پروسه انجام کارها در یک سازمان می‌باشد .

3. JDBC ^{۳۳}

استانداردی است که برای ارتباط با پایگاه داده‌ها توسط برنامه‌هایی که در بستر J2EE نوشته می‌شوند مورد استفاده قرار می‌گیرد .

4. JMS ^{۳۴}

برای تبادل پیغام‌ها میان برنامه‌های کاربردی در بستر J2EE

5. Java Mail

استانداردی برای ارسال و دریافت پیام‌های ایمیل (Email) توسط برنامه‌های کاربردی

31. java server pages
32. Enterprise Java Beans
33. Java Data Base Connectivity
34. Java Message Service

تکنولوژی servlet :

Servlet ها راه کاری هستند که شرکت Sun برای تولید برنامه‌های پویای سمت سرور در وب معرفی کرده است . ظرف نگهداری برنامه‌های servlet , jsp

Servlet / Jsp container

برای این که برنامه‌های servlet , jsp بتوانند اجرا شوند نیاز به محیطی دارند که این محیط شرایط اجرای چنین برنامه‌هایی را فراهم می‌کنند . ظرف نگهداری برنامه‌های servlet , jsp در واقع نرم افزاری است که این شرایط را برای صفحات jsp , servlet فراهم می‌آورد .

م : Apache Tomcat

ملزومات برنامه نویسی سمت سرور با استفاده از

servlet ها :

1. بسته نرم افزاری JDK بر روی سیستم نصب شود :
JDK مجموعه ای از قطعات و یا برنامه‌های نوشته شده به زبان java است که به عنوان پایه و مبنای سایر برنامه‌هایی که می‌خواهیم با استفاده از java ایجاد کنیم می‌باشد می‌توانیم آنرا به صورت رایگان از سایت java.sun.com دانلود کنیم .

(همانند .Net Framwork در تکنولوژی .Net .)

2. یک ظرف صفحات servlet / jsp مناسب بر روی سیستم نصب کنید .

برای این کار می‌توانیم از نرم افزار Apache Tomcat استفاده نماییم و این نرم افزار را هم می‌توان به صورت رایگان از اینترنت دانلود کرد .

3. نوشتن کدهای برنامه نویسی و یا ایجاد صفحات jsp با استفاده از یک ویرایشگر ساده و یا ویرایشگرهای مثل j Developer , j Builder و سپس کامپایل کردن کدهای servlet ها .

4. کپی کردن کدهای کامپایل شده و صفحات jsp در دایرکتوری مناسبی که توسط web container مشخص شده است و سپس اجرا کردن web container برای پاسخ گویی به درخواستهای client ها .

توضیحات جزئی تر در مورد تکنولوژی Servlet :

Servlet در زبان java کلاس های معمولی هستند که از کلاس های

http servlet به ارث می‌برند .

مکانیزم هایی از قبیل دریافت اطلاعات فرم های وب ، مدیریت و کنترل کوکی ها ، کار با سر آیند های پروتکل http و ... در این تکنولوژی پیاده سازی شده اند .

دو متد به نام do Post , do Get در داخل کلاس servlet که تعریف کرده ایم باید وجود داشته باشد .

کدهایی که صفحات html پویا تولید می‌کنند در داخل این متدها قرار می‌گیرند .

صفحات jsp :

در صفحات jsp کد های زبان java به همراه کد های زبان html در داخل صفحه وب قرار می‌گیرد راهکار jsp

تقریبا مشابه تکنولوژی asp است با این تفاوت که در jsp از یک زبان کامل به نام java برای نوشتن برنامه های سمت سرور استفاده می کند

این صفحات در پشت صحنه به صورت اتوماتیک توسط web container به صفحات سرولت تبدیل می شوند سپس این سرولتها کامپایل شده ، اجرا شده و خروجی به کاربر تحویل می شود .

اشیاء پیش فرض تعریف شده در صفحات jsp :

همانند صفحات ASP در صفحات jsp نیز اشیایی به صورت پیش فرض تعریف شده اند که می توانید از آنها استفاده کنید. این اشیا عبارتند از :

1. request همانند شیء تعریف شده در صفحات asp

2. response همانند شیء response تعریف شده در asp

3. session همانند شیء session تعریف شده در asp

4. page با استفاده از این object می توانیم به

پارامترهایی در ارتباط با صفحه jsp دسترسی داشته باشیم .

5. Out با استفاده از این شیء می توانیم رشته ای را در خروجی تحویل داده شده به Client چاپ کنیم .

نحوه نوشتن کدهای برنامه نویسی در داخل صفحات jsp:

1. <% کدهای برنامه نویسی %>

مثال :

<body>

<%

```

For ( int I =1 ; i<= 5 ; i++ )
{
String k = string . value of (i) ;
Out.println (“<h”+k+ “align =‘center’> Hello </h ”+k + k ” >”);
}
%>
</body>

```

2. <% = کدهای برنامه نویسی %>

برای ایجاد توضیحات در داخل صفحات jsp توضیحات مورد نظر خود را به صورت زیر قرار می دهیم :

```
<% _____ توضیحات _____ %>
```

3. <% code %> : برای یک session تعریف متغیر به صورت global است

تعریف متغیرها در صفحه به گونه ای است که در کل صفحه قابل دسترسی باشد

```
<%! String name = “nadir” ; %>
```

```
<html>
```

```
<body>
```

```
<% =name %>
```

```
</body>
```

```
</html>
```

زبان java :

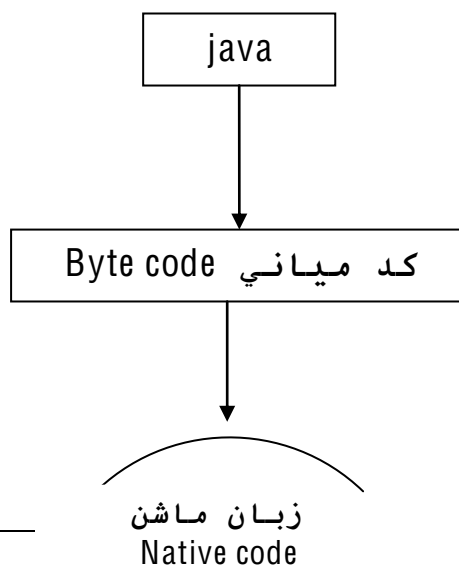
زبان java یک زبان شیء گرا در زبان اصلی بستر J2EE می باشد بنابراین در تکنولوژی های مرتبط با J2EE برای برنامه نویسی از زبان java استفاده می

شود . به عنوان مثال در صفحات jsp براي نوشتن کدها از زبان java استفاده مي کنيم .

! خصوصيت بارز زبان java مستقل بودن برنامه هاي نوشته شده توسط اين زبان از بستري است که اين برنامه ها مي خواهند بر روي آن بستر اجرا شوند

کدهاي زبان java در زمان کامپايل شدن به جاي تبديل به کد ماشين به کدهاي تحت عنوان byte code (معادل MSIL در .Net است) تبديل مي شوند . اين بايت کد مستقل از يک بستر خاص بوده (منظور از بستر ، بستر سخت افزاري و نرم افزاري است) و در هنگام اجرا توسط نرم افزاري به نام JVM^{۳۰} به زبان ماشين مختص آن بستر ترجمه مي شود .

معماري زبان JAVA :



مفهوم مستقل از بستر بودن برنامه‌های جاوا^{۳۶}

منظور این است که برنامه‌های نوشته شده با استفاده از جاوا زمانی که بر روی بستر خاصی نوشته شوند قابلیت اجرا در بسترهای دیگر را بدون هیچ تغییری در برنامه خواهند داشت

تفاوت زبان جاوا و زبان Javascript :

1. زبان جاوا یک زبان کامل است و بی زبان Javascript

یک زبان اسکریپتی با قابلیت‌های محدود است .

2. از زبان جاوا می‌توانیم برای نوشتن انواع

مختلف برنامه‌ها شامل برنامه‌های تحت وب ،

برنامه‌های ابزارهای موبایل ، برنامه‌های

رومیزی مبتنی بر ویندوز (برنامه‌های

دسکتاپ) ، برنامه‌های توزیع شده در سازمان و

غیره استفاده کرد در صورتی که از زبان Javascript

اکثراً برای نوشتن برنامه‌هایی تحت وب در سمت

client و به ندرت در سمت سرور استفاده می‌شود .

3. زبان Javascript تقریباً می‌توان گفت تنها در نام

ساختارهای برنامه‌نویسی مشابه زبان جاوا

است و هر دوی این‌ها توسط شرکت sun پشتیبانی

می‌شود .

محیط‌های توسعه برنامه‌های جاوا :

• Netspace (SUN)

• JBuilder (Borland)

• Jdeveloper (oracle)

• Eclipse (به صورت Opensource)

• IntelliJ IDEA (شرکت IBM)

مقایسه تکنولوژی‌های .NET و j2EE :

1: زبان برنامه‌نویسی مورد استفاده در j2 زبان java است در حالی که در .NET می‌توان از زبانهای برنامه‌نویسی مختلفی مانند VB , C++ , C# , J# , Pascal ...

2: برنامه‌های نوشته شده برای بستر j2EE با یک زبان مستقل از ماشین به نام بایت کد ترجمه می‌شود و برنامه‌های نوشته شده در .NET به کدهای MSIL ترجمه می‌شود.

3: بایت کدها در j2EE توسط jvm در هنگام اجرا به زبان ماشین تبدیل می‌شود این کار در .NET توسط CLR^{۳۷} بر روی MSIL^{۳۸} انجام می‌گیرد.

4: برنامه‌های کاربردی نوشته شده در j2EE مستقل از بستر هستند اما برنامه‌های نوشته شده در .NET تنها در بستر ویندوز قابل اجرا هستند.

5: برای اجرای برنامه‌های j2EE باید بر روی JDK^{۳۹} را نصب کنیم.

و برای اجرای برنامه‌های .NET باید .NET frame work را بر روی سیستم نصب کنیم.

6: به سبب مستقل از بسته بودن تکنولوژی j2EE توسعه دهندگان مختلفی در سراسر دنیا ابزارهای

37. Common Langvage Runtime

38. Microsoft intemodiate language

39. java Development Kit

متنوعی برای تولید برنامه‌های تحت j2EE ایجاد کرده‌اند. محیط‌های توسعه‌ای IDE^{۴۰} (یکپارچه) مانند JBuilder از شرکت Borland, J Developer از شرکت oracle. برای توسعه برنامه‌های j2EE وجود دارد. در حالی که ابزارهای مرتبط با تکنولوژی NET. معمولا توسط خود شرکت مایکروسافت توسعه می‌یابد مانند:

Visual studio .NET

7: تکنولوژی که j2EE برای نوشتن برنامه‌های تحت وب معرفی می‌کند تکنولوژی سرولت jsp است در مقابل NET. تکنولوژی ASP.NET را معرفی کرده است.

8: در برنامه‌های j2EE از JDBC^{۴۱} برای ارتباط با پایگاه داده‌ها استفاده می‌شود در NET. از تکنولوژی ADO.NET برای این کار استفاده می‌شود.

9: برای اجرای برنامه‌های تحت وب در NET. حتما باید IIS^{۴۲} را بر روی ویندوز نصب شده داشته باشیم.

برای اجرای برنامه‌های تحت وب در j2EE از نرم افزارهای متنوعی مانند Apache tomcat, Web sphere و... استفاده می‌کنیم.

10: هر دو تکنولوژی NET و j2EE ابزارهای مناسب و متنوعی برای توسعه برنامه کاربردی مورد استفاده در سازمان‌ها که به صورت توزیع شده با هم در ارتباط هستند, ارائه می‌کند.

: JSF

40. Integrated development Enviroment

41. Java Data Base Commenctivity

42. Internet information server

اگر چه صفحات ASP یک مکانیزم بسیار قوی و قابل انعطاف برای توسعه برنامه های تحت وب ارائه می دهد ملی این تکنولوژی مشکلاتی در ارتباط با توسعه نرم افزار های تحت وب مبتنی بر قطعات نرم افزاری دارد . در توسعه نرم افزار مبتنی بر قطعه هدف این است که قطعات نرم افزاری مستقل از همدیگر با رعایت استاندارد های تدوین شده توسط افراد و شرکت های مختلف ایجاد شوند . و سپس بتوانیم از این قطعات نرم افزاری ایجاد شده به صورت مستقل و بدون وابستگی به سایر قطعات و احیاناً در سایر قطعات نرم افزاری برای ایجاد کت برنامه تحت وب استفاده کنیم .

اگرچه jsp راه حل هایی همچون ایجاد تگ های سفارشی توسط کاربران را میسر می سازد ولی مفهوم کامل توسعه نرم افزار مبتنی بر قطعه را پشتیبانی نمی کرد بنابر این تکنولوژی JSF برای این منظور معرفی شده است . این تکنولوژی خود مبتنی بر JSP و servlet است در قیاس با تکنولوژی ASP این راه کار یعنی تکنولوژی JSF را می توانیم شبیه تکنولوژی ASP.NET در نظر بگیریم .

: Applet

Applet ها تکنولوژی های سمت client هستند برای این که داخل مرور گر اجرا شوند مرورگر بایستی از جاوا پشتیبانی کند . در یک Applet تقریباً از تمامی قابلیت های زبان برنامه نویسی جاوا می توانیم استفاده کنیم مثلاً فرم هایی همانند فرم های ویندوز داشته باشیم و دکمه هایی که با کاربر د تعامل می کند داشته باشیم

قرار دان يك Applet در داخل يك صفحه وب :

```

<body>
  <applet عرض="height", طول="width", آدرس كلاس اپلت
code = "
  <param مقدار پارامتر="value نام پارامتر
name = "
  <param مقدار پارامتر="value نام پارامتر ...
name = "
  <param مقدار پارامتر="value نام پارامتر
name = "
</applet>
</body>

```

جدول مقایسه ای تکنولوژی های مرتبط با وب :

سایر	.NET	javaEE	
/ Python / ایفل Php / perl / c	C# / vb / c++ / Delphi / ...	java	زبان برنامه نویسی
✓	*	✓	استقلال از سکوی اجرا
Perl / CGI / php / coldfusion	Asp.net / asp	Jsp / servlet / jsf	تکنولوژی برنامه نویسی سمت سرور در وب
Vbscript , javascript	Vbscript (Microsoft.co)	Applet / javascript(sun) ها	تکنولوژی برنامه

			نویسی سمت client در وب
Eclips و محیط ایفل و انواع محیط های دیگر	Vb.net	Netbeans / jbuilder / jdeveloper / elips / inteljideas	محیط های توسعه برنامه های کاربردی

: XML

تاریخچه زبان XML:

زبان sgml به عنوان پدر زبان XML شناخته می‌شود. زبان sgml^{۴۳} در واقع يك استاندارد بین‌المللی برای تعریف و توصیف ساختار و محتوای مستندات متنی است. ورژن اول زبان XML توسط کونسرسیم جهانی وب (W3c.org)^{۴۴} در سال 1998 معرفی شد.

معرفی زبان XML:

این اصطلاح مخفف extensible mark up language که زبان نشانه‌گذاری توسعه یافته است. در واقع نسخه ساده شده‌ای از زبان sgml است که شامل نسخه‌ای پر استفاده زبان sgml است. زبان XML در واقع زبانی است که برای نشانه‌گذاری داده‌ها مورد استفاده قرار می‌گیرد به طوری که بتوان این داده‌ها را در بسته وب و شبکه به آسانی منتقل نموده و این داده‌ها مستقل از بسته‌های نرم‌افزاری مختلف قابل خواندن و استفاده باشد.

زبان XML از بعد نشانه‌گذاری بسیار منعطفتر از زبان HTML است.

ویژگی‌های زبان XML:

1: اسناد XML به راحتی توسط انسان می‌تواند خوانده شود و مفهوم آن اسناد به راحتی قابل درک است.

43. Standard generalized Mark up language

44. world wide web consortium.org

2: اسناد XML به راحتی می‌تواند توسط کامپیوتر خوانده شده به صورت اتوماتیک تفسیر شود.

3: نشانه‌گذاری در زبان XML قابلیت انعطاف و تنوع زیادی دارد. به عبارت دیگر می‌توان گفت: هر نوع نشانه‌ای را می‌توان در زبان XML قرار داد.

4: خمیر مایه تشکیل دهنده اسناد XML متون ساده‌ای هستند که به راحتی توسط برنامه‌های مختلف در بسته‌های متنوع می‌تواند مورد استفاده قرار گیرد و به سبب ساده بودن آن به راحتی میان عناصر شبکه یا گره‌های شبکه جابه‌جا و منتقل شوند.

کاربردهای زبان XML:

به طور کلی زمانی که بحث انتقال داده در محیط شبکه مطرح می‌شود، XML به عنوان یک فرمت استاندارد جهت این تبادلات گزینه مناسبی است. موارد زیر نمونه‌هایی هستند که امروزه XML در آنها مورد استفاده قرار می‌گیرد:

1: سرویس‌های وب، (web service).

2: پروتکل‌های ارتباطی موجود در شبکه.

3: پایگاه داده و تکنولوژی‌های مرتبط با آنها.

4: فایل‌های پیکربندی نرم‌افزارهای کاربردی.

5: نمایش داده‌ها توسط صفحات html.

مکانیزم‌های تشخیص مجاز یا غیرمجاز بودن یک درخواست:

استفاده از:

1: آدرس IP

2: آدرس (URL) Domain

ساختار مستندات XML:

در ساختار اصلی در مستندات XML وجود دارد.

1: عناصر^{۴۵}:

عناصر در صفحات XML مانند تگهای html هستند که به عنوان مثال به صورت زیر مشخص می‌شوند:

```
<book> </book>
```

این تگها به صورت زوج وجود دارند. هر عنصر در XML یا به عبارتی هر tag در XML بیانگر توصیف کننده متنی است که آن را احاطه نموده است.

2: خصوصیات^{۴۶}:

توصیفاتی هستند که داده‌های احاطه شده توسط عناصر (Element) ها را بیشتر توصیف می‌کنند. خصوصیات در XML به صورت زیر مشخص می‌شود:

```
<element attribute = "value">
```

همان‌طور که در بالا مشخص شد خصوصیات عناصر به صورت مقدار= نام در داخل همان عنصر و قبل از علامت > مشخص می‌شود.

همان‌طور که می‌بینید چنین تعاریفی را در زبان html نیز داشتیم. مثلاً:

```
<img="pic.jpg"/>
```

قوانین نوشتن مستندات XML:

1: هر مستند XML با یک نشانه به صورت زیر آغاز می‌شود

45. Element
46. Attribute

<?xml version = "1.0" encoding = "utf-8" ?>

2: هر مستند XML داري نشانه اي است كه به عنوان نشانه اصلي تمامي مستند را در بر مي گيرد

3: هر عنصرى كه tag باز شدن آن وجود داشته باشد, tag بسته شدن آن هم بايد وجود داشته باشد.

4: اگر عناصرى داشته باشيم كه داراي محتوا نباشند و تنها داراي خصوصيت هستند بايد به صورت زير آنها را استفاده كنيم:

<img="pic.jpg"/>

اين tag محتوا ندارد.

5: ساختار سند XML بايد well-Formed باشد.

منظور از well-Formed اين است كه اگر عناصر يا tagهاي تو در تو در داخل سند XML استفاده مي كنيم, ترتيب باز و بسته شدن اين tag هاي تو در تو بايد رعايت شود, يعنى ابتدا tag بسته آخرين tag كه باز شده است, سپس tag بسته tag پدرش و... .

6: هر نشانه مي تواند شامل داده يا نشانه هاي ديگر باشد

7: هر نشانه مي تواند داراي صفاتي باشد كه به صورت زير در داخل نشانه آغازين قرار مي گيرد
نام صفت = مقدار صفت

8: محدوديتي در نامگذاري اسم نشانه ها در زبان XML به صورت ذاتي وجود ندارد

9: کاراکتر های خاص را که معنای خاصی در زبان XML دارند در صورتی که جزو داده های نشانه گذاری باشد باید به صورت کد شده در داخل مستند مشخص کنید

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"e	“	quotation mark

نکته: زبان XHTML زبان HTML ای است که استاندارد های زبان XML در آن رعایت می شود .

10: توضیحات در داخل مستند XML را داخل دو عبارت به صورت زیر قرار می دهیم

<!--

Comments

-->

تعیین قالب مستندات XML:

مستندات XML هیچ محدودیتی غیر از قوانین اخیر ذکر شده ندارد اما در مواردی لازم است که قالب و محدودیتی برای مستندات XML تعیین کنیم تا بتوانیم بر اساس آن ها اعتبار مستندات XML را که تولید می شود از ارزیابی نماییم برای این منظور از دو شیوه یا به عبارت دقیق تر از دو زبان استفاده می کنیم :

• DTD (Document Type Define)

• XSD (Xml Schema Defination)

مشتقات زبان XML :

به سبب قابلیت توسعه پذیری بسیار بالای زبان XML مشتقاتی با کاربرد های مختلف برای این زبان ایجاد شده است که برخی از آنها به شرح زیر است :

- 1 - XSL : که شامل XSLT و XSL-FO می باشد و زبانی است برای انتقال (توصیف) . قوانین انتقال دادهای مستندات XML به قالب ها و فرمت های دیگر .
- 2 - XPath : که استاندارد برای آدرس دهی گره ها در زبان XML است .
- 3 - XQuery : استاندارد برای تعریف پرسو جو ها در یک مستند XML است
- 4 - XPointer : استاندارد برای تعریف اشاره گر ها در یک مستند XML است
- 5 - SVG : استاندارد برای توصیف اشکال گرافیکی مبتنی بر زبان XML است که مانند فن آوری flash . تصاویر در این زبان مبتنی بر بردار است
- 6 - SMIL : استاندارد برای توصیف انیمیشن توسط XML می باشد .

نکته : XForms تعریف استاندارد جدیدی بر زبان XML برای ایجاد فرم های صفحات وب است که امکاناتی نظیر اعتبار سنجی داخل فرم ها به صورت پیش فرض در داخل آن ها پیاده سازی شده است .