

Fastcampus Sprint - Programming

Day 3. Network

Do it your self!

Numguess

- 1부터 100까지 정수 중 하나를 `answer` 라는 변수에 할당
- 사용자로 부터 임의의 값 하나를 받아 `guess` 라는 변수에 할당
- `answer` 와 `guess` 를 비교하여 정답여부를 출력

numguess

```
import random

answer = random.randint(1,100)
print(answer)
```

numguess

```
username = input("Hi there, What's your name?? ")
guess = eval(input("Hi, "+ username + "guess the number: "))

if guess == answer:
    print("Correct! The answer was ", str(answer))
else:
    print("That's not what I wanted!! The answer was ", str(answer))
```

Iteration

For, while

```
for 변수 in (리스트 or 문자열):  
    실행문1  
    ...
```

```
for i in ["python", "java", "golang"]:  
    print(i)
```

For, while

```
sum = 0
for i in range(1,11):
    sum += i
    sum = sum + i
    print(sum)
```

For, while

```
while 조건:  
    실행문1  
    ...
```

```
while name != "foo bar":  
    name = input("What's your name? ")  
    print("Hi, " + name + "So, where is foo bar?")
```

```
while 1:  
    print("Hello world!")
```


Iterations with Conditional Statements

Fizzbuzz

1부터 100까지 **반복하면서**,

3의 배수 = "Fizz"

5의 배수 = "Buzz"

15의 배수 = "FizzBuzz"

나머지 = 그 숫자

Fizzbuzz

```
num = eval(input("type the number: "))  
  
for i in range(1, num + 1):  
    if i % 15 == 0:  
        print("fizzbuzz")  
    elif i % 3 == 0:  
        print("fizz")  
    elif i % 5 == 0:  
        print("buzz")  
    else:  
        print(i)
```

Refactoring numguess

```
import random

answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
    guess = eval(input("Hi "+ username + ", guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    else:
        print("That's not what I wanted!! Try again!!")
```

give a hint!!

```
import random

answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")

while True:
    guess = eval(input("Hi, "+ username + "guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        print("Too high!! Try again!!")
    elif guess < answer:
        print("Too Low!! Try again!!")
```

```
import random

answer = random.randint(1,100)
username = input("Hi there, What's your name?? ")
trial = 5
while trial:
    guess = eval(input("Hi, "+ username + ". guess the number: "))

    if guess == answer:
        print("Correct! The answer was ", str(answer))
        break
    elif guess > answer:
        trial -= 1
        print("Too high!! Try again!!(%d times left)" % (trial))
    elif guess < answer:
        trial -= 1
        print("Too Low!! Try again!!(%d times left)" % (trial))
    if trial == 0:
        print("You are Wrong! The answer was ", str(answer))
```

Index

Network

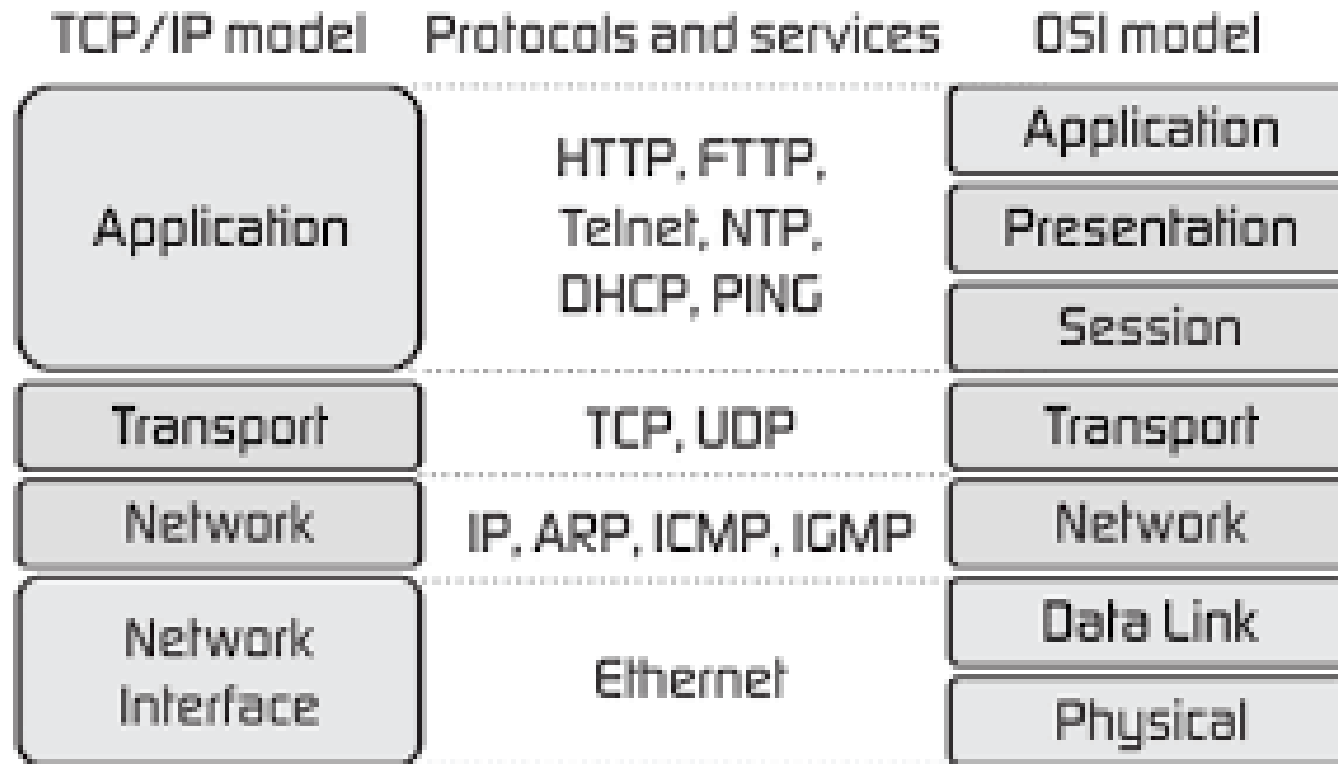
A computer network or data network is a telecommunications network which allows nodes to share resources.

--> 컴퓨터간 리소스를 공유 가능하게 만드는 통신망

Ethernet

- 전세계의 사무실이나 가정에서 일반적으로 사용되는 유선 LAN에서 가장 많이 활용되는 기술 규격
- ether == 에테르 == 빛의 매질
- IEEE 802.3 규약 기반
- OSI 7 Layer에서 Data-link Layer 에 위치

OSI 7 layer

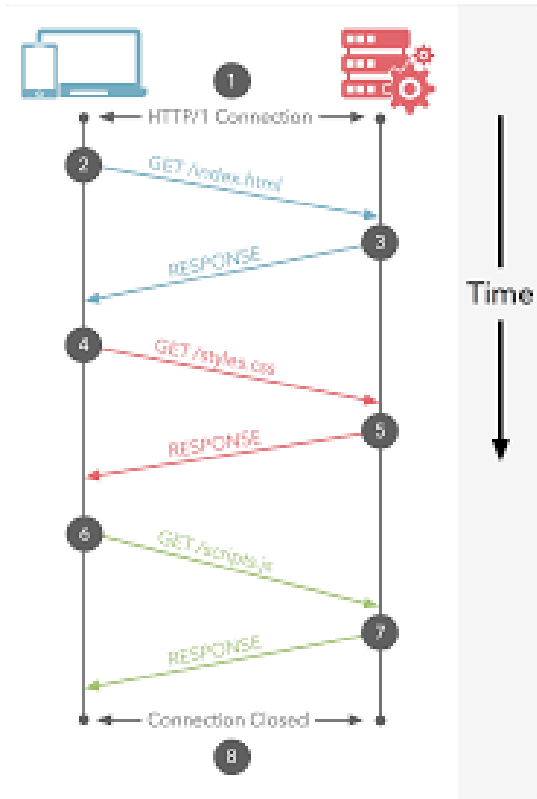


Internet

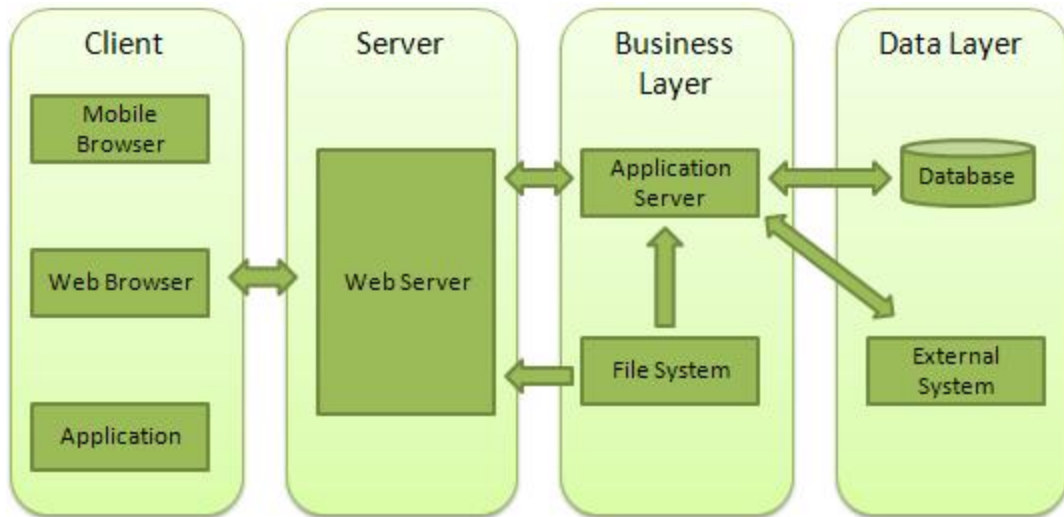
TCP/IP를 활용하여 정보를 주고 받는 통신 네트워크(www)

WWW(World Wide Web) == hypertext transfer through TCP/IP

Request & Response



Web Architecture



웹 개발 패턴의 변화

```
<html>
<head></head>
<body>
<h1>Static Header</h1>
<div>Static Contents</div>
</body>
</html>
```

- 1991 ~ 1999: Sir Timothy John "Tim" Berners-Lee가 하이퍼텍스트 기반의 프로젝트를 제안한 이후 정적인 콘텐츠를 중심으로 한 웹 기술이 발달

웹 개발 패턴의 변화

```
<html>
<head></head>
<body>
<h1>{% Dynamic Header %}</h1>
<div>{% Dynamic Contents %}</div>
</body>
</html>
```

- 1999 ~ 2009: Linux, Apache, Mysql, Php 중심의 동적인 서버, 정적인 클라이언트 모델이 지속됨

웹 개발 패턴의 변화

```
<html>
<head>
<script src="https://unpkg.com/vue"></script>
</head>
<body>
<h1>{{ header }}</h1>
<div id="app">
  {{ message }}
</div>
<script>
var app = new Vue({
  el: '#app',
  data: {
    message: '안녕하세요 Vue!'
  }
})
</script>
</body>
</html>
```

- 2010 ~ 현재: JavaScript!! (Dynamic Web Client)

HTML

- HyperText Markup Language

HTML

```
<!doctype html>
```

HTML

```
<!doctype html>  
<html>  
  <head></head>  
  <body></body>  
</html>
```

HTML

```
<head>  
  <meta charset="utf-8">  
  <meta name="viewport"  
    content="width=device-width, initial-scale=1.0">  
  <title></title>  
</head>
```

HTML

```
<body>
  <div id="main-wrapper">
    <h1 class="article-title"></h1>
    <p>This is <span>Home</span>.</p>
    <a href="#" target="_blank">hypertext</a>
    
  </div>
</body>
```

HTML - Semantic Element

```
<header>  
  <nav></nav>  
</header>  
  
<section>  
  <article></article>  
</section>  
  
<aside></aside>  
<footer></footer>
```

CSS

- Cascading Style Sheet
- 웹 문서의 스타일링을 위한 스타일시트

CSS basic style

```
body {  
    background-color: gray;  
}
```


CSS Selector

id, class, just tags

```
#some-id {color:#ff0000;}  
.some-class {color:#00ff00;}  
body {background-color:#dddddd;}
```

group selector

```
h1, h2, h3, h4, h5, h6 {font-family:Helvetica;}
```

child selector

```
body > h1 {align:center;}
```

attribute selector

```
p[title='introduce'] {font-family:Helvetica;}
```

JavaScript

- 객체 기반의 스크립트 프로그래밍 언어
- 웹페이지의 동적인 제어 목적
- Netscape의 Brendan Eich가 모카(Mocha)를 개발
- LiveScript -> JavaScript로 개명

JS DOM API

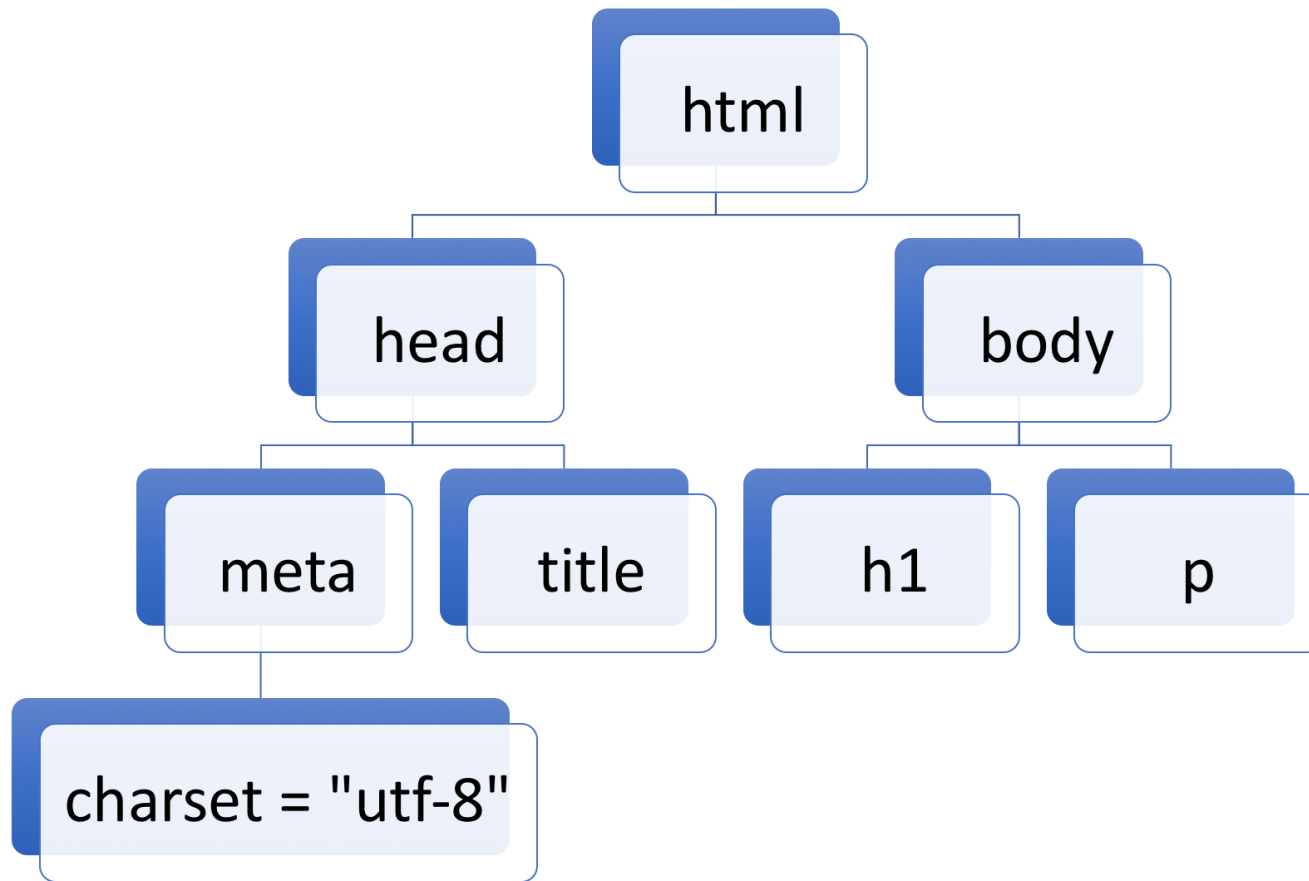
- DOM: Document Object Model
- HTML 문서를 분석하여 구조화

DOM

- Document Object Model

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My page</title>
  </head>
  <body>
    <h1>Home</h1>
    <p>Hello there!</p>
  </body>
</html>
```

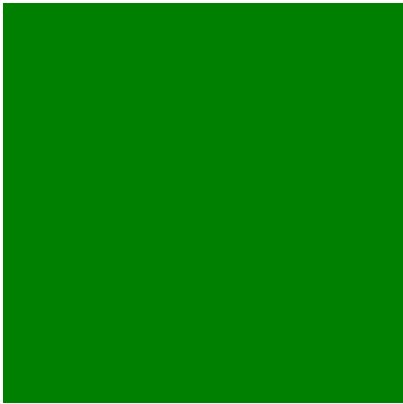
DOM



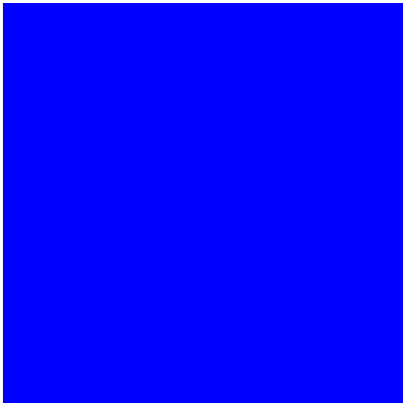
Static Web site - 1



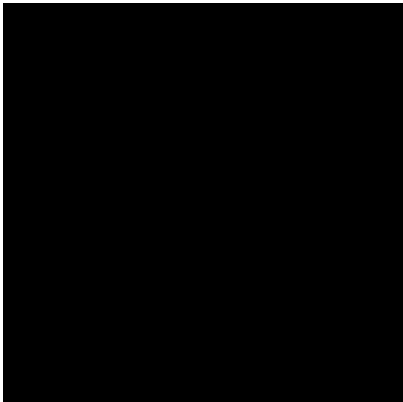
Static Web site - 2



Static Web site - 3



Dynamic Web site



Red

Green

Blue

Java != javaScript

Java	vs	javaScript
Sun	개발	Brendan Eich
JVM	구동방식	Script Engine(Browser)
C	영향	C
붕어	Like	붕어빵

XPath

- XML Path Language
- XML 문서의 요소와 속성을 통해 특정한 요소로 접근할 수 있도록 도와줌

Basic XPath

```
<body>
  <div id="site-wrapper">
    <h1 class="main-title">Page Title</h1>
    <div>
      <p class="paragraph">
        I am
        <span>a</span>
        boy.
      </p>
      <a href="#">Hypertext</a>
    </div>
  </div>
</body>
```

Basic XPath

`h1` : nodename

`/html` : root node

`//div` : select from current node

`.` : current node

`..` : parent node

`@` : attribute

Basic XPath

```
//body/div/p
```

```
//*[@id="site-wrapper"]/div/a
```

```
//*[@class="paragraph"]/text()
```