WEB UI BASIC

Javascript 1탄

NHN NEXT 우재우

본 자료는 NHN NEXT 윤지수 교수님의 자료를 바탕으로 제작했습니다.

아기다리 고기다리던

Javascript를 배워봅시다!

오늘은 기본만 쉽게 해보죠

일단 Javascript는

Java와 관계가 없어요ㅠ

(근데 자바지기는 JS(재성)입니다ㅋㅋㅋ)

쉽게 공부하려면 쉽고,

어렵게 공부하려면 끝도 없어요ㅋ

JS는 컴파일 언어가 아닙니다!

파이썬 같은

인터프리터 언어입니다~

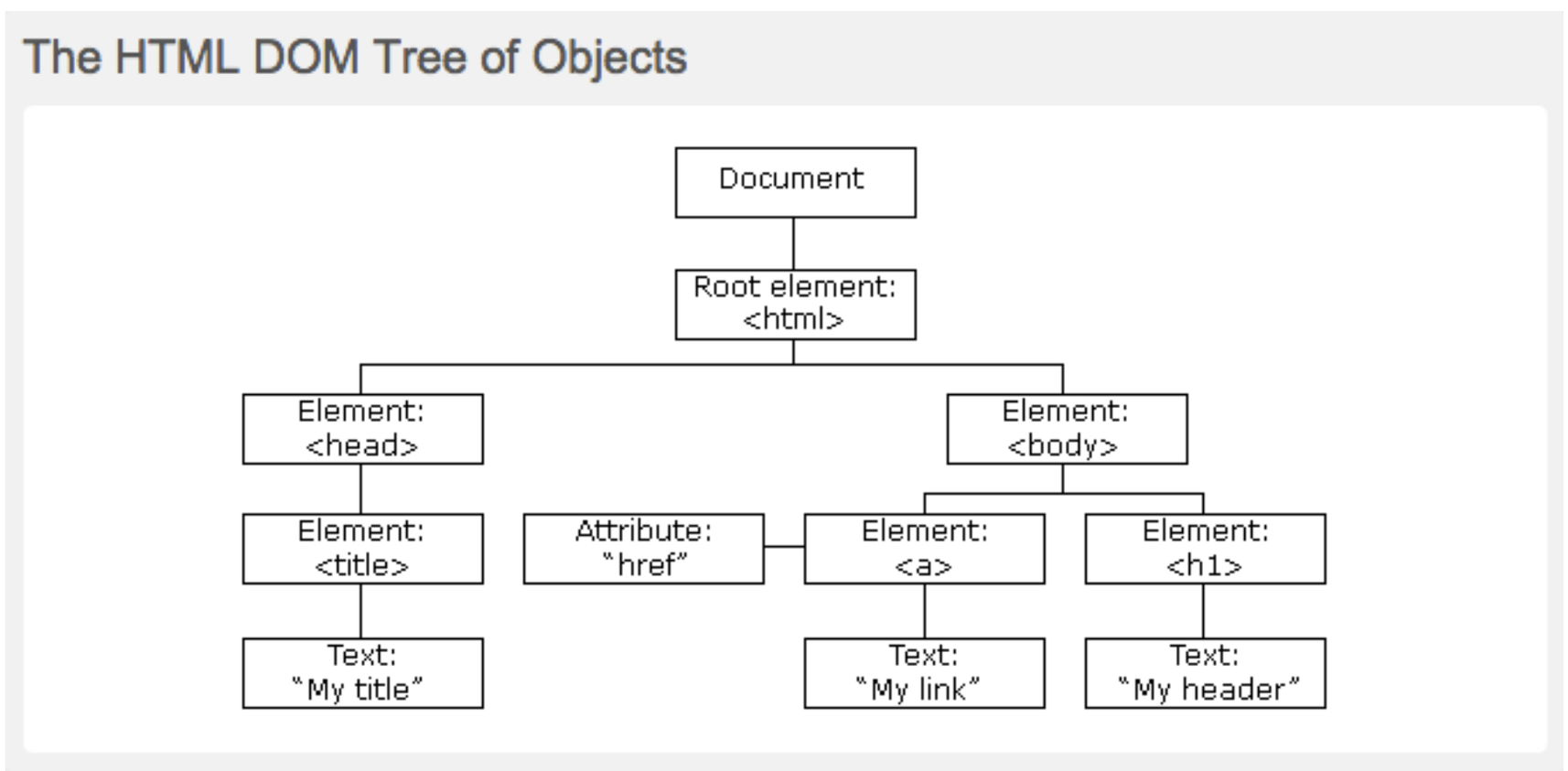
그래서 짜봐(Java)스크립트죠ㅋ

DOM(document object model)이라는게 있어요

HTML이 tree 구조로 만들어진다고 전에 말씀 드린거 기억나나요?

element 하나하나가 tree의 node가 되고,

그 node를 쉽게 찾을 수 있는 API가 DOM입니다!!!



출처: http://www.w3schools.com/js/js_htmldom.asp

그래서 JS가 하는 일은!!!

HTML의 구조를 조작합니다.(DOM 변경)

CSS를 동적으로 변경합니다.

사용자가 키보드/마우스 이벤트를 발생하면 처리해줍니다.

클라이언트 측 데이터를 변경/검증 할 수 있습니다.

서버에 요청을 보내고 받아서 처리할 수 있습니다.

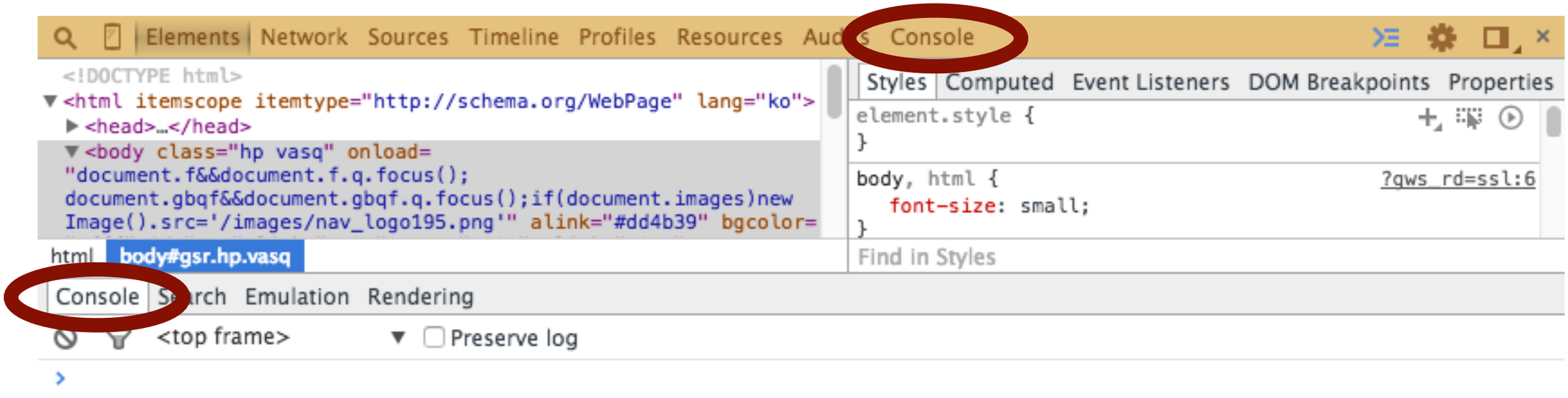
아, 심지어 서버도 만들 수 있어요!!! (Node.js)

Javascript 맛보기

개발자 도구에서 console 탭에 바로 코드를 쓸 수 있어요!

alert(“ hello ”);를 써봅시다!

줄바꿈은 shift + enter



Javascript를 적용하는 법 1 - internal 방식

html <script>라는 태그에 JS코드를 바로 씁니다!

```
<script>
    alert( “ Hello, world! ” );
</script>
```

[참고] HTML4에서는 <script type= ‘ text/javascript ’ >
라고 써주는데 HTML5부터는 <script>만 써도 됩니다.

Javascript를 적용하는 법 2 – external 방식

js파일을 따로 작성한 뒤 css external처럼 불러옵니다.

[test.js]

```
alert( “ Hello, world! ” );
```

```
<script src= ‘ test.js ’ ></script>
```

<script>의 위치는 어디가 좋을까?

```
<!DOCTYPE html>
<html>
<head>
   <여기?>
</head>
<body>
   <여기?>
   <코드들...>
   <여기?>
</body>
</html>
```

제가 좋아하는 www.html5doctor.com에서

실제로 어디있는지 살펴봅시다!

왜 하필 거기죠??

우리가 보는 웹페이지는 싱글뜨레드로 돌아갑니다!

쉽게 말해서 뭐 하나 하면 딴짓 못하는 스타일

그래서 브라우져가 JS를 읽는 동안은 다른 일 못 합니다ㅠ

JS가 뒤에 있으면 HTML, CSS 다 그리고 나서 다운받고 읽겠죠?

사용자 입장에서 생각해보면 흰화면에서 기다리는 것보다는

뭐라도 먼저 살짝 보이는게 반응속도가 빨라 보이겠죠?

위치를 꼭, 반드시, 머스트 지키지는 않아도 됩니다만

한 번쯤은 구글에서 이 이슈에 대해 검색해 보시길...

JS Data type

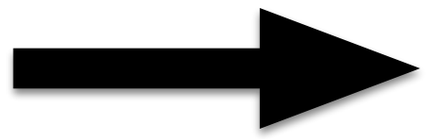
Number

String

Boolean

Array

Object



var

var는 척하면 척척척!

```
var num = 3;

var str = “hello”;

var bool = true;

var arr = [1, 2, “3”];

var obj = {“Hello”: “Hi”, “Bye”: ”See you soon”};
```

심지어 var를 안 써도 된다?!

test = 3; 이라고 쓰시고, 그냥 test;를 입력해봅시다!

type에 엄격한 언어들(예를 들면 C)을 쓰다가

JS를 보면 이게 뭔가 싶습니다 ㅋㅋㅋ

printf(); => console.log();

넌 type이 뭐니?

변수의 타입이 궁금할 땐 typeof!

Dynamic typing

```
var a;
console.log('typeof a is now,'+ typeof(a));
a = null;
console.log('typeof a is now,'+ typeof(a));
a = 1;
console.log('typeof a is now,'+ typeof(a));
a = "string";
console.log('typeof a is now,'+ typeof(a));
a = false;
console.log('typeof a is now,'+ typeof(a));
a = [];
console.log('typeof a is now,'+ typeof(a));
a = {};
console.log('typeof a is now,'+ typeof(a));
```

이거 허용해도 되는 건가요?

다음 값들이 어떻게 나오는지 예상해봅시다.

```
var result;

result = 30 + 30;

result = “result is” + 30;

result = “30” - 3;

result = “30” * 3;

result = -“30”;

result = “What is this?” - 30;
```

진짜 어처구니가 없죠? ㅋㅋㅋ

난 NaN이야...

NaN은 Not a Number 입니다.

대개 말도 안 되는 숫자 연산의 결과값으로 쓰입니다.

예를 들면

“ hello ” - 30

1 / “ smith ”

NaN == NaN

1 / 0 —-> 얘는 뭐가 나올까요???

내가 없는게 없는게 아니야

undefined vs null

undefined vs null

개발자도구 콘솔창에서 아래 코드를 입력해 봅시다!

```
var a;

a;
```

```
var b = null;

b;
```

undefined vs null

undefined는 변수를 선언만 했지 자료형이 결정되지 않은 상태!

null은 선언된 변수에 null이라는 빈값이 들어간 경우!

여기서 잠깐 쉬는 시간 ㅋ

Operators #1: Assignment operators

```
a = a + 3;        —->        a += 3;

a = a - 3;        —->        a -= 3;

a = a * 3;        —->        a *= 3;

a = a / 3;        —->        a /= 3;

a = a % 3;        —->        a %= 3;
```

Operators #2: Arithmetic operators

```
var a = 3;

a++;

a—;


var b = false;

b++;
```

Operators #3: Bitwise operators

```
15 & 9

> 9                 // 1111 & 1001 = 1001


15 | 9

> 15                // 1111 | 1001 = 1111


9 << 2

> 36                // 1001   —->   100100
```

Operators #4: Logical operators

```
true && true;
“hello” || false;
```

Operators #5: Comparison operators

```
“30” == 30;
true == “1”;

100 >= 99;
100 <= 99;
```

여기까지는 다른 언어랑 다를게 없군!

그런데!

JS에는 == 말고도

===이 있다!!!

Operators #5: Comparison operators

```
“30” === 30;

true === “1”;


null == undefined;

null === undefined;
```

===은 정확한 타입 비교까지 가능!

결론적으로 JS 타입 비교에서는

===이 안전하겠죠? :)

Operators #6: Conditional operators

```
var height = “182cm”;

var nHeight = parseInt(height);


var handSome = (nHeight > 180) ? true : false;

console.log(handSome);   // ?
```

if문

```
var a = 10;
if (a == 9) {
   a++;
} else if (a == 10) {
   a—;
} else {
   a = “Error”;
}
console.log(a);
```

우리가 알던 그 if가 맞습니다!

for문

```
var result = 0;

for (var i = 1; i <= 10; i++) {
    result += i;
}

console.log(result);
```

while, do while, break, continue, switch 등

원래 알던 것들 거의 똑같이 작동해요~

자세한 것은 각자 검색해서 공부하는 걸로...

미션!!!

다음 배열에서 짝수의 개수가 몇 개인지 출력하세요!

for를 이용하며, 문자열이 나오면 loop를 탈출합니다.

(참고: Array에는 length property가 있어요 ==> arr.length)

```
var arr = [1, 2, 3, 4, 23, 43, 57, 100, “ 333 ”, 10];
```

우리는 JS맛만 봤다!!!

겸손해야 합니다 ㅠ

아직 갈 길이 구만구천구십구리

이번 주는 딱히 대단한 숙제가 없어요!!!

신나죠?!?!

GOOD 프로젝트 계속 발전시킵시다ㅎ

다음 주도 JS 계속!!!