JAVAScript

1.수업소개

Html은 정적임

But, 사용자와 동적으로 상호작용을 하기 위해 발달

HTML+JavaScript

# 2.수업의 목적

Input type=”button” value=”[~]” onclick=”[JS]”

HTML은 스스로 변화하지 못한다.

JS로 조정 가능

# 3.HTML과 JS의 만남 : script 태그

자바스크립트라는 것을 표시하게 위해 <script> 태그 사용

<script>

document.write(1+1)

</script>

# 4.HTML과 JS의 만남 : 이벤트

<input type="button" value="hi" onclick="alert('hi')">

10~20개정도의 event

<input type="text" onkeydown="myFunction()">

# 5.HTML과 JS의 만남 (콘솔)

간단하게 시험할 때 검사의 console

~.length

검사 element에서 esc누르면 console이 따로 뜸

# 6.데이터타입 - 문자열과 숫자

Javascript의 data type

* [기본 자료형 (Primitive)](https://developer.mozilla.org/ko/docs/Glossary/Primitive) 인 여섯가지 데이터 타입
  + [Boolean](https://developer.mozilla.org/ko/docs/Glossary/Boolean)
  + [Null](https://developer.mozilla.org/ko/docs/Glossary/Null)
  + [Undefined](https://developer.mozilla.org/ko/docs/Glossary/undefined)
  + [Number (en-US)](https://developer.mozilla.org/en-US/docs/Glossary/Number)
  + [String](https://developer.mozilla.org/ko/docs/Glossary/String)
  + [Symbol](https://developer.mozilla.org/ko/docs/Glossary/Symbol) (ECMAScript 6 에 추가됨)
* 별도로 [Object](https://developer.mozilla.org/ko/docs/Glossary/Object) 도 있음

대 소문자 구분해야함

문자열은 ‘’로 감싸줘야함

# 7.변수와 대입 연산자

Variable vs constant

변수 선언할땐 var 을 선언해줘

장점:

단점:

8.웹브라우저 제어

# 9. CSS 기초 : style 속성

<h2 style="color:powderblue ;background-color: brown">JavaScript</h2>

# 12.제어할 태그 선택하기

document.querySelector(‘’)

검색으로 찾아

“”안에는 ‘’, ‘’안에는 “”

# 13.프로그램,프로그래밍,프로그래머

HTML:컴퓨터 언어(시간 순서 상관X)

JS:컴퓨터 언어이면서 프로그래밍 언어(조건, 순서에 따라 실행되야함)

# 14. 조건문 예고

‘===

<는 문법기호이기 때문에 &lt; less than

>는 &gt; greater than

# 18.리팩토링 중복의 제거

리팩토링: 동작은 그대로지만 코드를 효율적으로 바꾸는 것

본인 참조 -> this로 바꿀 수 있음 (id도 지움)

Var V1 = 으로 중복요소 제거

19. 반복문 예고

# 20.배열

<https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array/Array>

# 21.반복문

While문 짤 때는 언제 끝날것인가를 생각하면서 짜야함

document.write('<li>'+coworkers[i]+'</li>');

데이터에따라 바뀌는 로직말고 유연하게 다 동작하는 로직을 짜자

document.write(<li><a href="http://a.com/"></a></li>);

22.배열과 반복문

# 23.배열과 반복문의 활용

검색도 프로그래밍의 일부

console.log(~);

내 페이지를 1억명이 본다면 밤 샐 가치가 있다.

# 24.함수예고

function nightDayHandler(){}

# 25.함수

함수 또는 매서드

보통 CSS의 <style>은 <head>에, JS의 <script>는 <body>에?

함수는 입력과 출력이 중요

26.매개변수와 인자

Parameter & argument

# 27.함수 (리턴)

Return

함수 결과를 이용하여 활용할때를 위해

# 28.함수의 활용

함수 선언할땐 인자로 self

함수 사용할땐 매개변수로 this

# 29.객체 예고

정리정돈의 수단(중복 제거)

함수명 안겹치게끔 조심

파일이 안복잡하게 폴더에 넣는 것

객체에 속해있는 함수 = 매서드

순서없는 배열?

# 31.객체와 반복문

Var coworkers={

“programmer” : “egoing”,

“designer”: “leezche”

}

Coworkers[“data scientist”]: “taeho”

Coworkers.bookkeeper = “duru”

For(var key in coworkers) {

Document.write(key+’:’+coworkwes[key]+’<br>’);

}

32.객체프로퍼티와 메소드

객체에 소속된 함수: 메소드

객체에 소속된 변수: 프로퍼티

coworkers.showAll=function(){

}

coworkers.showAll

var Body = {

setColor: function(color){

document.querySelector('body').style.color=color;

},

setBackgroundColor: function(color){

document.querySelector('body').style.backgroundColor=color;

}

}

함수: 정리정돈의 도구

객체: 함수와 변수의 정리정돈의 도구

# 34.파일로 쪼개서 정리 정돈하기

<script src="colors.js"></script>

웹 서버입장에서는 나쁜일 (접속이 늘어남)

But, 캐시입장에서는 효율적

# 35.라이브러리와 프레임워크

다른 사람이 만든 소프트웨어를 갖다씀

Library vs framework

Library :내가 만드려는 프로그램의 부품들이 모여 있는 곳(땡겨다 쓰는 느낌)

Framework: 우리가 만들고자 하는 것에 따라 공통적으로 필요한 반제품적인 것(들어가서 만드는 느낌)

JS: jQuery 생산성이 훨씬 높아짐, 직관적

Javascript를 이용해 누군가 만들어 놓음

 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

$('a') = 모든 a를 jquery로 제어하겠다

$('a').css('color',color)

# 36.UI vs API

UI: User Interface(유저가 프로그래머가 만든 것을 이용하는 것)

API: Application Programming Interface (프로그래머가 만들고 사용하는 것)

# 37.수업을 마치며

프로젝트 시작하자

최소한의 도구들만 가지고 시작하고 문제에 직면하고 해결해봐라(점점 넓어짐)

Document -> DOM -> window

Ajax: reload없이 동작을 보고 싶을 때

Cookie:reload되어도 현재상태를 유지하고 싶을 때, 사용자를 위한 개인적인 서비스 제공 가능

Offline web application: 인터넷이 없어도 동작

webRTC: 화상통신 web app

speech: 사용자의 음성으로 정보 전달

webGL: 게임같은 3차원 그래픽으로

webVR: 가상 현실