**#1.0 Why JS?**

* JavaScript란 ? 웹에서 쓰이는 하나뿐인 프로그래밍 언어. 왜 하나뿐일까? 옵션이 하나뿐이기 때문! 웹에서 웹사이트를 만들고 interactive한 것을 만들고 싶을때 쓴다.
* 백엔드는 여러가지 언어가 있지만, 프론트엔드에서는 자바스크립트 필수임.
* 좋은 점은 fragmentation이 없다.(분열) 웹사이트를 하는 사람은 JS를 아니까 같은 언어를 이용하여 말할 수 있다.
* 단점도 존재한다. 갇힐 때도 있고 할수있는게 없을 때도 있다. 바꾸거나 업데이트, 원하는걸로 교체 못함.
* 그치만 장점은 웹은 빠르게 발전하기 때문에 JS도 빠르게 발전하고 막강해지고 영향력이 커지고 있다. 그리고 대체할 언어가 없다 !!

**#1.1 Super Powers of JS**

* 위치나 날씨 가져오기. 할일 목록 만들 수 있다. 실시간을 만들 수 있다.(실시간 채팅같은거). 휴대폰에서도 작동할 수 있는 앱 만들 수 있음.
* impact js : frame work 또는 library 같은거. 게임
* JS는 interactive한 모든것에 사용됨. 웹에서만 사용하는게 아님 !!

**#1.2 ES5, ES6 ES....WTF!?!?!**

* ECMAScrip : Specification의 명칭 같은것. ES5는 ECMAScript5를 말한다. 이건 Specification에 대한 업데이트이다. ES5 ES6는 그냥 Specification의 버젼을 나타냄.

예를들어) 구글 크롬이 ES6를 사용한다면 구글크롬 개발자들이 새로운 Specificationd을 시행했다고 볼 수 있는 것. 새로운 안내문을 발행한것과 같다.

* Specification이란 JS는 꽤나 중앙 집중화되어있어서 누군가 업데이트를 하면 모든 브라우저에서 작동을 하게 된다. JS는 specification인데 체계 매뉴얼(안내책자)같은 것이다. 이걸 읽고 이대로 적으면 이러한 창이 띄워집니다 ~ 와 같은. 설명문 같은거 !! 많은 사람들에게 검증 받은 !
* 브라우저(크롬, 인터넷 익스플로어, 사파리 등)는 이 specification을 받아서 자기들 방식으로 실행한다. 같은 결과를 이루려 노력하지만 각자 방식으로 실행함.
* 바닐라 자바스크립트는 JS의 한 종류로 library가 없는 것을 이야기한다. 화장이 없는 JS 도움이 없는 JS 날것의 JS를 말한다.

**#1.3 VanillaJS**

* 업데이트를 할 수 없는 오래된 버젼 같은 경우에는 우리가 그것을 모두 바꿀 수 없다. 180도 다르거나 모던하게 바꿔줄 수 없다. 만약 그럼 엄청나게 많은 인터넷 사이트들이 망가진다(엑박이 뜸)
* 바닐라 자바스크립트는 브라우저를 통해 우리에게 제공된 JS라고 보면된다. 온갖 임시방편과 지저분한 것들.(항상 사람들이 바닐라 JS에 대해 언급하는거임.)
* 어떤 사람은 바닐라 대신 커피 스크립트(JS의 섹시버전)를 배우고, 어떤 사람은 JS를 엄청 현대적으로 바꾸는 법을 배우기도하고, 어떤 사람은 묶는 법을 배운다.
* library나 framework는 메이크업, 화장 같은 존재. js를 원래 모습보다 좀 더 예쁘게 만들어주는것. 배우기 쉬움.
* 바닐라 JS는 어렵기때문에 먼저 배우면 좋다. (framewor만 잘 다루는 사람 -> 포토샵 잘하는 사람이지만 사진못찍는 사람 떠는 색감에 대한 센스 없는사람)
* 바닐라 js를 배우는 것은 웹에서의 기초가 되는 언어를 배우는 것과 같다.

**#1.4 Hello World with Javascript**

* JS파일은 항상 body 안의 맨 아래에 있어야한다. 제일 마지막에 추가를 해야함.

( <body> <h1></h1> <다른태그 ></다른태그>

<script src=”index.js”></script> <body> )

js 코드를 원할경우에 스크립트 태그 안에 적어두면 됨.

* alert(‘blabla’); --> 경고문 ! JS에서의 hello world 같은거임.
* console.log --> 콘솔창. (괄호안에 쉼표 넣어서 여러가지 넣을 수 있다.) 안에 메세지 없으면 실행 안됨.
* html css에 JS파일을 추가하는건 매우 쉽고 심플하다

이유는 ? 웹 브라우저에는 JS가 설치되어 있기 떄문.

**#1.6 Your first JS Variable(변수!)**

* variable : 변경되거나, 변경될 수 있는 것.

ex) a = 2 // a는 바뀔 수 있다. 존재하지 않았지만 우리가 존재하게 만들었고 바꿀 수 있다. 이게 변수임.

* 프로그래밍 언어는 일반 언어처럼 따라야하는 규칙과 문법이 있다
* JS는 우리가 나쁜것을 하는 것을 허용한다는 문제점이 있다. (자식을 훈육하지 않는 아빠처럼)

예) 변수를 이상하게 쓰더라도 JS는 그것을 최대한 이해하기 위해 노력한다. 실행하기위해 최선을 다한다. 심지어 그게 좋은 문법이 아니더라도

* 따라서 우리가 문법을 철저하게 지켜서 만들려면, 몇가지를 주의하면된다 !
* 모든 instruction은 각기 다른 expressions(lines)에 있다. 모든 expressions은 한줄에 있어야 한다. 모든 instructions도 한 줄에 있어야 하듯이. 한 expression이 끝나는 곳이 선언하는 방법은 ; 이다.(instructions를 제외하고)
* 변수를 만들고 싶을때는 이렇게 작동한다.

create - initialize - use 생성하고 초기화하고 사용한다 (필요할때는 생성과 초기화를 동시에한다. 그리고 나중에 사용 (예, a=221 a를 생성하여 221로 초기화한거임))

* 규칙은 변수를 초기화하기 전에 이 단어를 넣어야함.

예) let a =221; 처럼. let b = a-5;

변수 초기화 하거나 생성할때에는 let을 써야하지만 이미 생성된 것을 사용할 때에는 안적어줘두됨. 단지 참조만 하면됨.

만약 let 안쓰고 그냥 쓰면 이미 초기화하고 사용하는 것처럼 작동한다. 이건 별로임. 안좋음.

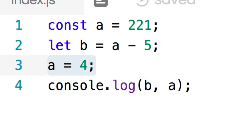
텍스트이(가) 표시된 사진

자동 생성된 설명여기서 a=4는 let안쓰고 다시 선언 안했음 그저 업데이트 해준거임.

**#1.7 let, const, var**

* 변수를 지정해주고 나중에 또 그 변수이름을 사용하는 것을 방지하기 위해 const 사용한다.

const : constant 상수(안정적)라는 뜻. 변하지 않는다. (기본으로 쓰자 ! let은 정말 필요할때만 쓰도록 !)

이렇게 하면 상수변수에 대입했다는)에러생김

* 에러를 읽고 시도하는것은 중요함. 모든 정보나 대부분의 정보가 에러에 있기 때문 !!
* var : variable.

문제는 그렇게 많지 않지만 JS가 var을 적용한 이상한 규칙이 있음. 그리고 let을 쓴것처럼 var을 바꿀 수도 있음.

**#1.8 Data Types on JS**

* 코멘트 : 주석처리. 코드에 남길 수 있는 노트같은것. 주석처리 되지 않은 모든 코드는 프로그램이 실행하려한다.

‘// ‘ 쓰면 된다 !! 여러줄 하고 싶을 때는 /\* 2번째줄 3번째줄 \*/ 넣으면됨.

* string : 모든걸 같이 붙이는 것. 보통 스트링은 텍스트 스트링을 말한다.

const what = “Nicola” 하면 스트링이 저장됨 근데 여기서 “” 제거하면 프로그램은 nicolas라는 키워드를 어디선가 찾으려고 함. 그럼 오류나는거임 찾을수 없다고 ! 어딘가에 있는데 찾을 수 없다고 보는거임. 아직 존재하지 않으니까

* “ “ 안에 넣으면 문자 처리 됨.
* Boolean : True or Flase

이렇게 쓴다. const wat = false; 모두 소문자로 쓰고,””없이 쓴다. 이건 텍스트가 아니기 때문. 바이너리 세계(이진법) 에서는 모든것은 0또는 1이다.

* number : 숫자
* float : 소수

**#1.9 Organizing Data with Arrays**

* 데이터를 정렬하는 두가지 방법 ! 하나는 array 다른 하나는 object
* array : 데이터를 저장하는 곳인데, 리스트를 같이 저장하는것. 여러 string들을 하나로 묶는 것. (text, Boolean, number, 변수 . . 등 만 가능 섞어서도 가능)

예) const daysOfWeek = [ “ㅁ”, “ㅠ”, “ㅊ”]

* array안의 규칙 : dayOfWeek[2] 하면 거기 자리에 해당되는 글자 나온다. 0부터 시작이다. daysOfweek[1]= “ㅠ”임.
* 문법을 위한 규칙 !!

camel case : 변수명을 언제나 소문자로 시작해서 만약 변수명 중간에 스페이스가 필요하면 스페이스 대신에 다 음 단어 첫글자를 대문자로 써주는것.

\* 스페이스가 없더라도 변수명에 이상한 문자 넣으면 안됨

**#1.10 Organizing Data with Objects**

* object과 array의 다른점 : object에는 각 value에 이름을 줄 수 있다.

\* 리스트로 만들고 싶을때는 array를 정의. 간단하니까 그냥 리스트 정의하면됨

* 원하는 방식으로 정렬하고 싶다면 ? 각 값에 변수 지정해주고 싶다면 ?(name : blabal 같이) -----> object 선언
* object를 선언하기 위해서는 { 컬리 브라켓 }을 써주면된다.
* object는 array같이 동작하지 않는다. object는 실제 객체를 만드는 거임.

label을 내가 저장하고 싶은 data에 줄 수 있음.

텍스트이(가) 표시된 사진

자동 생성된 설명이렇게 !!

-> name은 텍스트 아니다. 변수니까 !! 그래서 큰따옴표로 감싸지 않는다.

-> gender에만 접근하는 방법 ---> nicoInfo.gender 라고 해주면 됨 !

* const안의 object 안의 값을 바꿀 수 있다.

nicoInfo.gender=”Female” 해서 보면 이렇게 나옴.

* 하 지 만 !! object 자체를 바꿀 수 없음.

예) nicoInfo = true 이렇게는 안된다눈거임.

* Object를 Array 안에 넣을 수 있다.

array를 object안에도 넣을 수 있다. favMovies : [“blabla”, “abc”] 이런식으로

텍스트이(가) 표시된 사진

자동 생성된 설명 이렇게도 가능

* , 빼먹지 말기 ! string 쓸때 “ 큰따옴포 “ 앞뒤로 넣는거 잊지 말기 !
* js가 말썽이어도 html과 css은 멀쩡하다.

**# 2.0 Your first JS Function**

* console.log(nicoInfo.favFood)

--> console은 object이다. (nicoInfo가 object인 것 같이)

--> .log 는 안의 내용 (.favFood 같이). 함수(function)이다. console object 안에 있는 함수라는 뜻

텍스트이(가) 표시된 사진

자동 생성된 설명console.log(console) 한 내용

* console.log, alert 등 곧 보게 될 많은 함수들은 내장함수이다
* 함수(function)란 어떤것의 기능이다. 어떤 걸 수행하려는 한 부분으로, 우리가 원하는 만큼 쓸 수 있다. (한 코드조각으로 우리가 원하는 만큼 쓸 수 있는것)
* 함수 만들기(JS 함수 정의하는 문법)
  + function 원하는함수이름(인자(변수))){ 실행 }
  + 예) function sayHello(x) {

console.log(‘Hello!’, x); }

* + 실행 : sayHelllo(“Nicolas”);
* 인자(argument) 즉 변수명은 아무이름이나 사용 가능하다 !! 함수 안에서 사용하게 될 이름이 되는거임.
* 인자(argument)는 변수 같은거인데, 우리가 주는 값을 저장한다.
* 두 개 이상을 입력받고 싶으면 콤마를 사용하면 된다. (변수 두개, 실행할때도 입력값이 두개)
  + 이게 외부에 있는 데이터를 읽는 함수를 만드는 방법임. 함수에게 외부에 있는 데이터를 주는 방법.
* console.log() 는 괄호안에 아무것도 안넣으면 실행 안된다

--> 즉, 여기에 뭐든 넣고 싶은 것을 두개의 따옴표 사이에 넣으면 함수는 그걸 사용한다는 말.

\* argument(인자, 변수)를 무한히 가질 수 있다 !!!!!!

**#2.1 More Function Fun**

* + JS에서는 텍스트를 써야하고 텍스트에 변수를 섞어 쓸 수 없다. 만약 우리가 많은 콤마를 쓰고 싶지 않다면 (예전에는 + 썼음)
  + 백틱(₩)(`) : 이건 큰따옴표나” 작은따옴표랑’ 매우 다르다.
* 예) function sayHello(변수, 변수) {

console.log(`Hello ${변수} you are ${변수} years old`); }

* returning X : console.log와 함께 컴퓨터 사이언스의 가장 기본이지만 저거랑 다르다.
* return : 돈을 내면 반드시 잔돈을 받아야 하는 것과 같은 개념



이렇게 console.log로 greetNicolas를 반환해보면 정의되지 않았다고 나온다.

“greetNicolas는 sayHello의 실행된 결과 값”인데 여기서는 아무것도 반환하지 않았다. 아무것도 안썼음.

‼️‼️‼️ 여기서 반환되게 하려면 함수(만든거) 안의 console.log -> return으로 바꿔주면 된다. ()괄호도 빼고

* 그렇게 바꾸고 console.log(맨 밑줄)로 찍어낸다. (이거 없으면 콘솔 창에 아무 것도 안나옴)
* console.log처럼 우리만의 객체(object) 만들기

텍스트이(가) 표시된 사진

자동 생성된 설명

\* calculator : object(객체)이다. console 처럼

\* plus : calculator 안의 인자를 가지는 함수(2개를 가짐) log 처럼

#2.2 JS DOM Functions

* 브라우저에서 우리에게 제공하는 더 많은 다른 함수들이 있다.
* 여기서 html을 다뤄서 JS와 함께 다루면 어떻게 될까 ????/
* CSS와 같이 JS에서도 element 선택 할 수 있다. 그래서 변경 할 수 있다. h1을 선택해서 바꿀수 있는 것과 같이.
* JS에서 id 선택하는 방법 : document.
* document도 object이다. document를 대표하는 html을 준다. html document는 JS가 되는거임
* document도 많은 .함수()를 가지고 있다..