JavaScript 정리

1. 정의 : ‘웹페이지에 생동감을 불어넣기 위해’만들어진 프로그래밍 언어
2. 구성 : 스크립트 언어 + 인터프리터 언어

스크립트 언어? 응용 소프트웨어를 제어하는 컴퓨터 프로그래밍 언어

(응용 소프트웨어?

운영 체제에서 실행되는 모든 소프트웨어 + os위에서 사용자가 직접 사용하게 되는 소프트웨어 =>시스템 소프트웨어의 부분(애플리케이션))

* 어플리케이션을 제어하는 프로그래밍 언어
* 자바스크립트로 작성한 프로그램을 스크립트라고 부름
* 스크립트는 명령어의 모음이다.

컴퓨터가 단계별로 따라 할 수 있는 명령들로 구성

상호작용 방식에 따라 필요한 부분만 사용

상황에 따라 코드의 서로 다른 부분 실행

인터프리터 언어? - JAVA랑 달리 보통의 문자 형태로 작성

고급언어로 작성된 프로그램들을 실행하는 두가지 방법

1. 프로그램 컴파일: 원시 코드를 기계어로 번역
2. 인터프리터에 통과시키는 방법
3. 소스 코드 직접 실행
4. 효율 적인 중간 코드로 변환 후, 바로 실행
5. 인터프리터 시스템의 일부인 컴파일러가 미리 만들어 둔  
   코드 호출
6. 자바스크립트언어로 브라우저에서 할 수 있는 일

브라우저 환경에선 웹페이지 조작, 클라이언트와 서버의 상호작용에 관한 모든 일

1. 페이지에 새로운 html추가, 기존 html 혹은 css 수정.
2. 마우스 클릭, 포인터의 움직임, 키보드 키 눌림 등에 반응
3. 네트워크를 통해 서버에 요청 + 파일 다운로드, 업로드

(AJAX 나 COMET의 기술사용)

1. 쿠키를 가져오거나 설정하기, 사용자에게 질문, 메시지 보여주기
2. 클라이언트 측에 데이터 저장하기(로컬 스토리지)
3. 자바스크립트언어로 브라우저에서 할 수 없는 일

\*보안!!! 때문에 제약이 존재함 -> 직접적으로 파일 다루기등

1. 자바스크립트만의 강점
2. 브라우저 인터페이스를 쉽게 만들 수 있다.

이유 : HTML/CSS 완전 통합 가능

간단한 일은 간단하게 처리

모든 주요 브라우저에서 지원, 무엇보다 기본언어임!

JavaScript 정리

**왜? 자바스크립트?**

: ‘동적인 웹페이지를 만들고 싶어서’

2. (1) Vscode + node.js설치

(2) 스크립트 파일 선언위치(head)에 선언

asyncs 와 defer

둘다 html이 parsing중 서버에서 js파일 다운.

Html이 parsing이 다 되면 js파일 실행.

차이? Asyncs는 파일이 여러 개라도 한번에 다운  
 but 크기에 따라 순서 상관없이 excute하는 시간이 달라질   
 수 있다.

Defer는 파일이 여러 개라도 한번에 다운  
 script순서에 따라 순차적으로 excute.

Defer을 많이 쓴다!! 가장 좋다!!!

1.변수(let) 와 상수(const)

변수, 상수 선언시

let: 변수를 쓸 때 쓴다. var는 쓰지마!!

Constants: 상수를 쓸 때. 변경되지 않는 datatype  
 보안상의 이유로 사용, 쓰레드들의 안정성, 실수를 줄임.

Data의 type

Primitive type: single item: number, string, boolean, undefined, null등

Object type: single item들을 하나로 묶어서 한 단위로  
 관리하기 위함 타입. Function, first-function등이 있다.

Function도 변수에 할당이 가능하다!!!!!!

Primitive type 중 DataType에 따라서.

undefined: 변수가 정의되지 않거나 값이 없을 때

number: 데이터 타입이 수일 때

string: 데이터 타입이 문자열일 때

boolean: 데이터 타입이 불리언일 때

object: 데이터 타입이 함수, 배열 등 객체일 때

Operator(연산자) 중 Comparisons(비교 연산자)

==(동등): type이 같이 않은 경우 자동 형 변환 후 비교.

같은 객체를 바라보고 있는지 판별.

!=(부등): type자동 변환 후 두 연산자가 같지 않은 경우 참.

메모리의 다른 객체를 바라보고 있는지 판별

!==(불일치): 두 연산자가 같지 않거나, 같은 자료형이 아닐 경우 참

===(일치): 자료형 변환 없이 엄격하게 같은지 비교.

>=: 왼쪽이 오른쪽보다 크기나 같으면 참

>: 왼쪽이 오른쪽보다 크면 참

<=: 왼쪽보다 오른쪽이 더 크거나 같은 경우 참

<: 왼쪽보다 오른쪽이 더 큰 경우 참

Operator(연산자) 중 Logical(논리 연산자)

Expr1&&Expr2: 인수 2개 중에 둘 다 참일 경우return true,

하나라도 false가 있으면 false반환

Expr1||Expr2: 인수 2개 중에 하나라도expr1 == true return expr1,

둘 다 false인 경우 false반환

!expr: 피연산자를 Boolean 타입으로 변환하여 true/false로 값을   
내고, 그 결과의 반대값을 반환

참고)\*api : Application Programming Interface

2. function

(1) 기본적인 빌딩 => 함수는 한가지 일을 하는 것으로 정의!!

function name(param){body…..return….}

1>함수 이름이 있는 경우

함수 선언 -> function hello(){console.log(‘hello’);}

함수 호출 -> hello();

cf) 함수의 param에 따라 다른 함수로 인식되기 때문에 형식,  
 개수 다 맞추어야 한다.

특징)  
 (1) 다른 변수와 마찬가지로 변수에 할당가능

(2) 파라미터로 전달이 되며 return 값으로도 return가능

(2) scope

비유로 기억하면 쉬움.

밖에서는 안이 보이지 않고, 안에서만 밖을 볼 수 있다.

(3) early return, early exit.

조건이 맞지 않는 것부터 설정하여 빨리 함수를 종료 시키는 방식

\*Arrow function

(anonymous function)

const simpleprint = (param) =>console.log(‘simplePrint!);

\*Callback function( non-block, 비동기 방식의 함수 )

다른 함수의 매개변수(param)로 함수를 전달하고,  
 어떠한 이벤트가 발생한 후 매개변수로 전달한 함수가 다시 호출되는 것

즉) 어떠한 일을 다른 객체에게 시키고, 그 일이 끝나는 것을   
기다리지 않고 끝나고 부를 때까지 다른 일을 하는 것을 말함

왜 사용? 프로세스를 차례대로 수행하기 위해서.

(2)  
function randomQuiz(answer,printYes,printNo){

if(answer === ‘love you’){

printYes();

} else {

printNo();

}

}

(3)

const printYes = function(){

console.log(‘yes’);

};

Const printNo = function print(){

console.log(‘no!’);

};

(1)

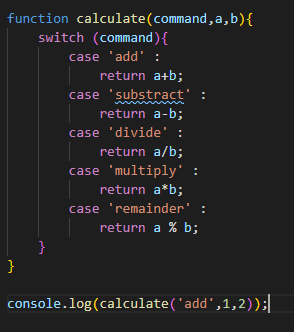
randomQuiz(‘wrong’, printYes, printNo);

rnadomQuiz(‘love you’, printYes, printNo);

함수 연습문제

function calculate(command,a,b)

command : add, substract, divide, multiply, remainder



3. Class

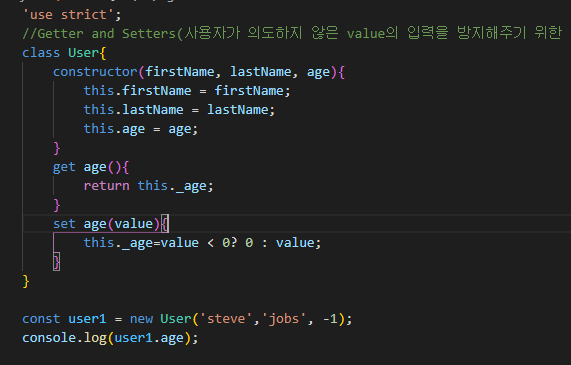
정의: template(설계도, 틀)

Object(key : value), 띄어쓰기 시(‘key’ : ‘value’)

클래스를 이용하여 새로운 인스턴스(메모리에 올림)를 생성하면

Object가 된다.

구성: 속성(fields) + 행동(method) \*method가 없는 경우도 있다.



object사용 방식



4. Array(배열)

배열은 동일한 타입의 값을 순차적으로 저장하는 용도로 사용.



배열에는 유용한 메서드들이 있다.

Array.from() -> 유사 인스턴스를 생성하는 메서드

Array.isArray() - >인자로 전달받은 객체가 배열이면 true리턴

<원본 배열을 수정하는 메서드>

1.배열의 처음 혹은 마지막에 요소를 추가하거나 제거

.push(): 마지막에 새로운 요소 추가. 변경된 길이 리턴

.pop(): 마지막 요소 제거, 제거한 요소 반환

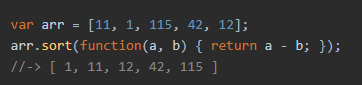
.unshift(): 첫번째 자리에 새로운 요소 추가, 변경된 길이 리턴

.shift(): 첫 번재 요소를 제거한 후, 제거한 요소 반환

.splice() : 배열의 특정 위치에 배열 요소를 추가하거나 삭제.  
 삭제한 요소 리턴

.reverse(): 배열 요소의 순서를 뒤집기.

.sort(): 배열 내부의 요소를 정렬하는데 사용(기본 오름차순)



비교함수를 사용. 두개의 인자(a,b)를 받으며 배열 요소의 가장  
 처음 두 요소부터 순차적으로 비교해가며 정렬을 실행.

5. Loop(반복문) – while, for, do

대표적으로 for과 while을 사용.

for문은 반복 횟수를 정할 수 있을 때,  
 while은 반복횟수를 정할 수 없을 때 사용.

while에서 break, continue 사용.

break;를 넣으면 반복문을 중단.(원하는 값을 찾고 빠져나올 때)

continue;를 넣으면 다음 반복문 실행.(수행작업에서 예외를 둘 때)

do~while(while의 변형)

내용을 먼저 실행한 후에 조건을 비교.  
 내용이 최소 한번은 무조건 실행.

let I = 0;

do{

alert(i);

i++;

} while{I < 10}

DOM Selectors

1)getElementsByTagName

요소의 element 태그명으로 선택

복수를 저장할 수 있기에 배열로 저장됨

2)getElementsByClassName

요소의 class 이름으로 선택

Class 특성 값이 지정된 값과 일치하는 모든 요소들을 선택

3)getElementById

id특성 값이 주어진 값과 일치하는 하나의 개별 요소를 선택

~요즘 대세!~

4)querySelector

Css 선택자 문법을 이용하여 하나 혹은 그 이상의 요소를 찾음.

일치하는 요소들 중 첫번째 요소만을 리턴

5)querySelectorAll

Css 선택자 문법을 이용하여 하나 혹은 그 이상의 요소를 찾음.

발견된 요소들을 모두 리턴

DOM 메서드가 하나 이상의 요소를 리턴할 때는 NodeList객체  
(요소 노도의 모음- 각 요소에는 index번호가 부여됨 – 순서는 html에 해당요소가 정의 된 순서대로-배열이 아니라 컬렉션이다.)를 리턴 한다.

1> NodeList는 요소 노도의 모음이다.

2> 각 요소에는 index번호가 부여된다.

3> 순서는 html에 해당요소가 정의된 순서대로 부여 된다.

4> 배열이 아니라 Collection타입의 객체이다.

length: 객체 내에 저장된 아이템의 개수를 리턴

item(): 괄호에 저장된 인덱스 번호에 해당하는 노드를 리턴

하지만 객체로부터 아이템을 조회할 때는 []사용이 보편적임.

사용법 선택자 + 루프 실행하기 + 아이템 조회)

var elements = document.getElementsByClassName(‘hot’);

if(elements.length >=1){

var firstItem = elements[0];

}

~선택자로 선택 후 특성을 검사하여 값 가져오기!~

6)getAttribute()

특성 값을 가져온다.

7)setAttribute()

특성에 값을 대입한다.

8)hasAttriubte()

요소 노드가 지정한 특성을 가지고 있는지 검사한다.

9)removeAttribute()

요소 노드에서 지정한 특성을 제거한다.

DOM 요소의 내용 가져오기/수정하기

요소의 내용을 조작하는 방식 2가지

1) 텍스트 노드까지 탐색하기

대상 요소가 오직 텍스트 노드만 가지고 있을 때 사용  
 nodeValue 노드가 가지고 있는 텍스트에 접근

2) 상위 요소 조작하기

요소의 텍스트 노드와 자식 요소들 모두에 접근이 가능.

요소가 텍스트 노드 및 그에 이웃하는 자식 요소들을 가지고

있는 경우에 적합

innerHTML: 텍스트와 마크업을 가져오거나 변경한다.

보안에 취약함.

textContext: 텍스트만 가져오거나 변경한다.

innerText: 텍스트만 가져오거나 변경한다.(권장X)

3) DOM조작으로 요소 추가하기.

1단계: createElement(): 새로운 요소 노드 생성

2단계: createTextNode(): 새로운 텍스트 노드를 생성

3단계: appendChild(): 위에 것들을 DOM트리에 추가

Ex) //새로운 요소, 텍스트 노드를 생성 후 변수에 저장

var newE1 = document.createElement(‘li’);

Var newText = document.createTextNode(‘퀴노아’)

//텍스트 노드를 새요소에 추가한다.

newE1.appendChild(newText);

//새 요소가 추가될 위치를 선정

var position = document.getElementsByTagName(‘ul’)[0];

//새로운 요소를 해당 위치에 추가한다.

position.appendChild(newE1);