**Js 강의 내용 정리**

#1

1.변수

let 변수 지정

ex)let hi = hi라는 변수 지정

변수는 변경할 수 있고 선언 후 값 지정 가능

ex)hi=안녕을 hi=안녕하세요로 바꾸면 변경됨

ex)let hi; hi=’안녕’ 가능

2.상수

Const 상수 지정

console.log(변수) = 가로 안에 변수 넣으면 console에 출력됨

상수는 변경할 수 없음, 선언 할 때 값 초기화해줘야함

ex)const hi =’안녕’

3.변수나 상수 규칙

-첫번째 글자는 영어, $, \_만 사용 가능

-첫 글자 이외에는 숫자 사용 가능

-영어 소문자와 대문자는 다른 것으로 구별 (보통 소문자-변수, 대문자-상수)

-공백 허용 X

4.Tip

우선 const로 선언 -> 값 변경이 필요한 경우 let 변경 추천함

#2

원시 타입 : 메모리에 변수를 위한 공간이 할당되고 그곳에 값이 저장됨

1. 원시 데이터 타입 : 객체가 아니면서 메서드도 가지지 않는 데이터

type of – 어떤 데이터 타입인지 알려줌

1) number

ex) const number = 123; console.log(typeof number); ->number

2) bigint : 숫자 데이터 타입으로 표현할 수 있는 범위를 넘는 아주 작은 숫자 혹은 아주 큰 숫자를 표현 (n 추가)

ex) const number = 123n; console.log(typeof number); -> bigint

3)string : 따옴표로 감싸진 문자열

ex) const name = ‘한지민’; console.log(typeof name); ->string

종류 : ‘(따옴표) . “(큰 따옴표) , `(백틱)

백틱 사용법 `` 안에 문자는 그대로 넣고 변수는 ${변수}으로 한 번에 표현 가능

4)Boolean : 참 또는 거짓 둘 중 하나만 값으로 가질 수 있는 논리 타입

ex) const isTure = 10 < 20; console.log(typeof isTrue) -> Boolean

5)null : 값이 존재하지 않음을 나타냄

6)undefined : 값이 지정되지 않음을 나타냄

ex) 변수를 선언 후 값 초기화하지 않았을 때 : let hello; console.log(hello);

#3

참조 타입 : 힙 heap이라는 별도의 공간에 저장하고 메모리에는 변수의 주소값을 저장해서 참조해서 사용함

1. 배열

ex) const arr = [1, 2, 3];

2. 객체

ex) const obj = {

name: ‘한지민’ -> name이라는 키에 한지민이라는 값

job: ‘개발자’

}

표기법 1) 점 표기법 ex) console.log(obj.name); -> 한지민

표기법 2) 대괄호 표기법 ex)console.log(obj[‘name’]) -> 한지민

3.배열과 객체의 중첩

ex) const arr = [1, 2, 3, [4, 5]]; console.log(arr);

ex) const arr = [1, 2, 3, { name: "한지민" }]; console.log(arr);

#4

1. 산술 연산자

2. 증감 연산자

++1 -> 1 증가

--1 -> 1 감소

3. 비교 연산자

== 추상 비교, === 엄격 비교 (보통 === 사용) -> true/false

!== 양변이 같지 않은지 비교 -> true/false

4. 논리 연산자

&& (and) : 양 변이 참이어야 true가 나옴

||(or)

! (not)

5. 삼항 연산자

형식 : 조건 ? 참일 때 실행될 부분 : 거짓일 때 실행될 부분

ex) console.log(2 < 3 ? "참" : "거짓");

6. 널리쉬 연산자 ?? : 값이 확정되어 있는 변수 찾음

ex) const e = undefined; const f = null; const g = '한지민' console.log(e ?? f ?? g); -> 값이 문자로 확정되어 있는 한지민 출력됨

7. 비트 연산자

&, |, ~, ^, <<, >>

8. 대입 연산자

+= -= \*= /= %= \*\*=

ex) number = number + 2; -> number += 2;로 간단하게 사용 가능

9. 전개 구문 (…)

전개 되어서 출력되는 것

ex) console.log(...numbers);

#5 조건문

1. if

If, else는 한 번만 사용 , else if는 if와 else 사이에만 사용 & 여러 개 사용

ex) const h = 10; const i = 20; const j = 20;

if (h > i) { 🡪 if가 거짓이니 건너뜀

console.log("h가 더 큽니다 !");

} else if (i === j) { 🡪 else if가 참이니 실행

console.log("b랑 c가 같습니다!");

} else {

console.log("여기는 언제 출력될까요 ?");

}

2. switch

switch (number) {

    case 1: 🡪 number가 1이면 실행

        console.log(number);

        break; 🡪 break는 case와 한 쌍으로 사용 ! break가 없다면 break를 만날 때까지 다음 코드 실행함

case 10:

        console.log(number);

        break;

    default: 🡪 아무것도 해당되지 않으면 실행됨, 맨 마지막에 사용

         console.log('아무것도 해당되지 않아요!')

}

#6 반복문

1. for

ex) for (let k = 0; -> 시작값 k < 5; -> 조건식 k++) {

  console.log(k);

}

\*break , continue

ex) for (let m = 0; m < 10; m++) {

  if (m === 7) {

    break; 🡪 7에서 멈춤 0~6까지 출력 continue; 🡪 7만 X, 0~6 + 8~9까지 출력

  }

  console.log(m);

}

2. for of

ex) const arr10 = [1, 2, 3];

for (const n of arr10) { 🡪 arr10에 있는 1,2,3이 n으로 들어가는 것

  console.log(n);

}

3. while

ex) let o = 0;

while (o < 10) {

  console.log(o++);

} 🡪 조건식이 참이면 실행되었다 다시 올라가서 반복해서 거짓이 될 때까지 실행

4. do while

ex) do {

  console.log(p++);

} while (p<10); 🡪 우선 실행 후 조건식 확인

#7

1. 함수

function 사용

ex)

function bok(main) {

console.log(`${main} 볶음밥`);

}

bok("새우");

bok("제육");

인자 = 함수의 입력 값 = arguments

매개변수 = 함수의 입력 변수 = parameter

나머지 매개변수 = rest parameter

입력한 인자의 개수가 부족할 때 -> 부족한 인자는 undefined로 처리

입력한 인자의 개수가 많을 때 -> 배열에 따라 알아서 할당되므로 무시됨

나머지 매개변수 사용 시

function print(a, b, ...rest) {

  console.log(a);

  console.log(b);

  console.log(rest);

}

print(10, 20, 30, 40, 50, 60, 70); 🡪 a,b에 10,20 들어가고 rest에 나머지 값들 들어감

2. 함수의 표현식

function sum(a,b) {

 console.log(a+b);

}

|

표현식으로 변경한 것

const sum function(a,b) {

    console.log(a+b);

}

3.화살표 함수

ex) const sum = (a,b) => console.log(a+b);

인자가 하나일 때는 괄호 생략 가능