자바스크립트를 배울때 도움되는 자료

MDN : https://developer.mozilla.org/ko/

모던 자바스크립트 튜토리얼 : ko.javascript.info

VSCode : 에디터

크롬 : 자바스크립트 코드를 실행하기 위한 브라우저

브라우저 없이 자바스크립트를 실행하기 위해서는 Node.js 를 설치하면 됨.

( ) 소괄호 : parentheses / { } 중괄호 : braces / [ ] 대괄호 : brackets

빽틱 ` : 빽틱 안에서는 줄바꿈을 할 수 있다

\n : 줄바꿈

백슬래시\ : 문자가 다르게 해석되게 한다(이스케이핑)

5e4 = 50000
5e-4 = 0.0004
0b : 2진법
0 : 8진법
0x : 16진법

typeof 로 자료형 확인

문자열 3.14를 숫자로
perseint('3.14') = 3 (문자열을 **정수**로)
persefloat('3.14') = 3.14 (문자열을 **소수**로)
Number('3.14') = 3.14 (문자열을 정수,소수 상관없이 **숫자**로-문자가 섞여있으면 **nan**)

prompt() : 문자열을 입력받는 창

perseint(prompt()) : 숫자를 입력받는 창

컴퓨터의 부동소수점 계산

문자열 크기는 사전순

특수문자도 크기비교가 가능하다
'&'.charCodeAt() : 해당 특수기호의 번호를 알려준다

!! 불값으로 형 변화했을때 false 가 되는 6가지 경우false / ''빈문자열 / 0 / Nan / undefind / null

빈값 자료형 undefind

조건문

switch문

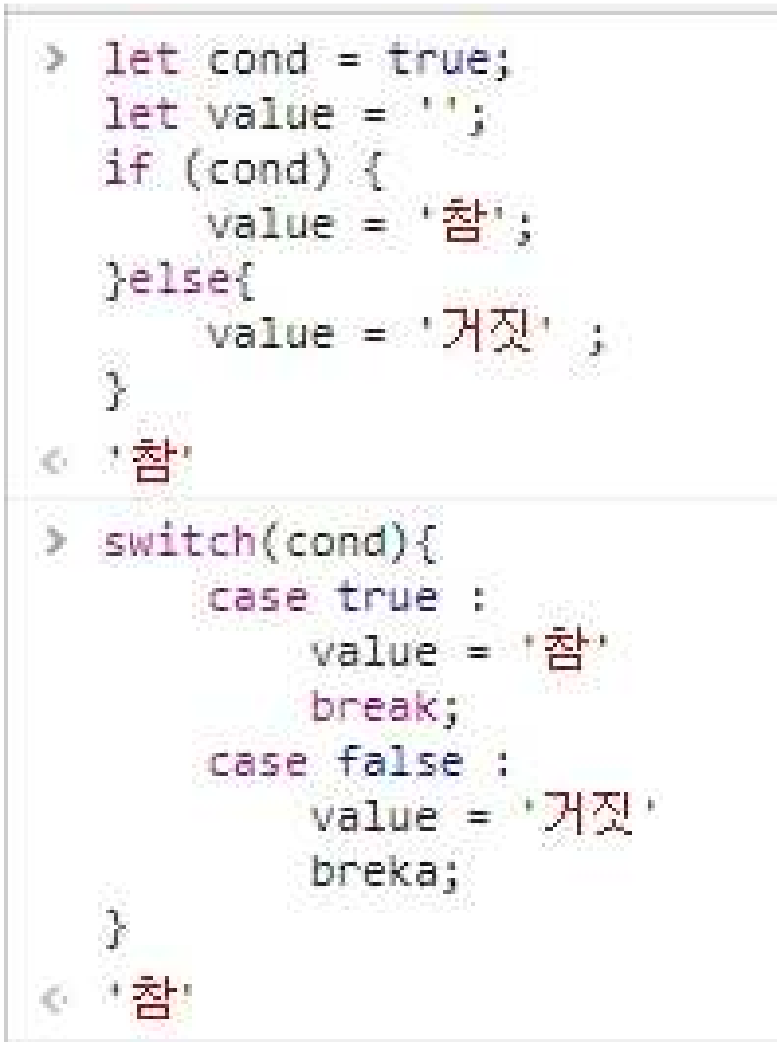
- 일치하는 case를 발견하면 그 아래 case가 모두 실행됨
- break으로 수동으로 case를 빠져 나올 수 있음
- default : 어떤한 case도 일치하지 않을때 실행, switch문의 상단, 하단 어디에도 사용 할 수 있음.

조건부 연산자 = 삼항 연산자

조건식 ? 참일때 실행되는 식 : 거짓일때 실행되는 식

중첩 조건부 연산은 다시한번 확인! p.104

p106. 1분퀴즈
if문을 switch문으로, 조건부 연산자로 바꿔보기



```
> let cond = true;
  let value = cond ? '참':'거짓';
< undefined
> value
< '참'
>
```

반복문

Shift+Esc : 브라우저 작업관리자

p112. 1분 퀴즈
1~100까지 반복문 for문으로 출력

```
> for (let i=0; i<100; i++){
      console.log(i+1);
  }
```

```
> for (let i = 0; i < 5; i++) {
    if (i % 2 === 0) continue;
    for (let j = 0; j < 5; j++) {
      if (j % 2 === 0) continue;
      for (let k = 0; k < 5; k++) {
        if (k % 2 === 0) continue;
        console.log(i, j, k);
      }
    }
  }

  // i==0 continue
  // i==1 j==0 continue
  // i==1 j==1 k==0 continue
  // i==1 j==1 k==1 console.log(1, 1, 1)
  // i==1 j==1 k==2 continue
  // i==1 j==1 k==3 console.log(1, 1, 3)
  // i==1 j==1 k==4 continue
  // i==1 j==1 k==5 조건X
  // i==1 j==2 continue
  // i==1 j==3 k==0 continue
  // i==1 j==3 k==1 console.log(1, 3, 1)
```

중첩 반복문

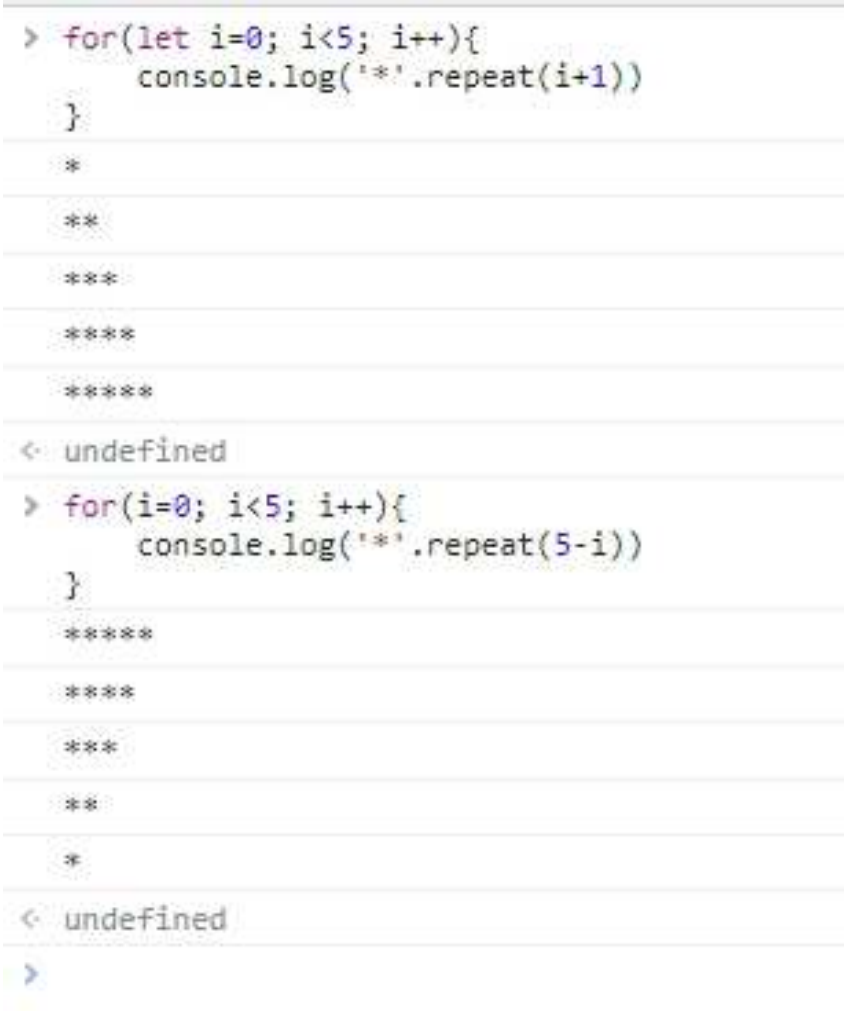
```
> for (let i = 0; i < 10; i++) {
    for (let j = 0; j < 10; j++) {
      console.log(i, j);
    }
  }

// i==0  j==0   console.log(0,0);
// i==0  j==1   console.log(0,1);
// i==0  j==2   console.log(0,2);
// i==0  j==9   console.log(0,9);
// i==0  j==10, 조건X
// i==1  j==0    console.log(1,0);
// i==1  j==9    console.log(1,9);
// i==1  j==10  조건 X
```

```
> for(let i=0; i<10; i++){
      if (i%2==0) continue;
      for(let j=0; j<10; j++){
          if (j%2==0) continue;
          console.log(i,j,i*j)
      }
  }
1 1 1
1 3 3
1 5 5
1 7 7
1 9 9
3 1 3
3 3 9
3 5 15
3 7 21
3 9 27
5 1 5
5 3 15
5 5 25
5 7 35
5 9 45
7 1 7
7 3 21
7 5 35
7 7 49
7 9 63
9 1 9
9 3 27
9 5 45
9 7 63
9 9 81
< undefined
>
```

```
> for(let i=0; i<10; i++){
      for(let j=0; j<10; j++){
          if (i*j%2==0) continue;
          console.log(i,j,i*j)
      }
  }
  1 1 1
  1 3 3
  1 5 5
  1 7 7
  1 9 9
  3 1 3
  3 3 9
  3 5 15
  3 7 21
  3 9 27
  5 1 5
  5 3 15
  5 5 25
  5 7 35
  5 9 45
  7 1 7
  7 3 21
  7 5 35
  7 7 49
  7 9 63
  9 1 9
  9 3 27
  9 5 45
  9 7 63
  9 9 81
< undefined
>
```

```
> for(let i=0; i<10; i++){
      for(let j=0; j<10; j++){
          if (i%2==0 || j%2==0) continue;
          console.log(i,j,i*j)
      }
  }
  1 1 1
  1 3 3
  1 5 5
  1 7 7
  1 9 9
  3 1 3
  3 3 9
  3 5 15
  3 7 21
  3 9 27
  5 1 5
  5 3 15
  5 5 25
  5 7 35
  5 9 45
  7 1 7
  7 3 21
  7 5 35
  7 7 49
  7 9 63
  9 1 9
  9 3 27
  9 5 45
  9 7 63
  9 9 81
< undefined
>
```

별찍기

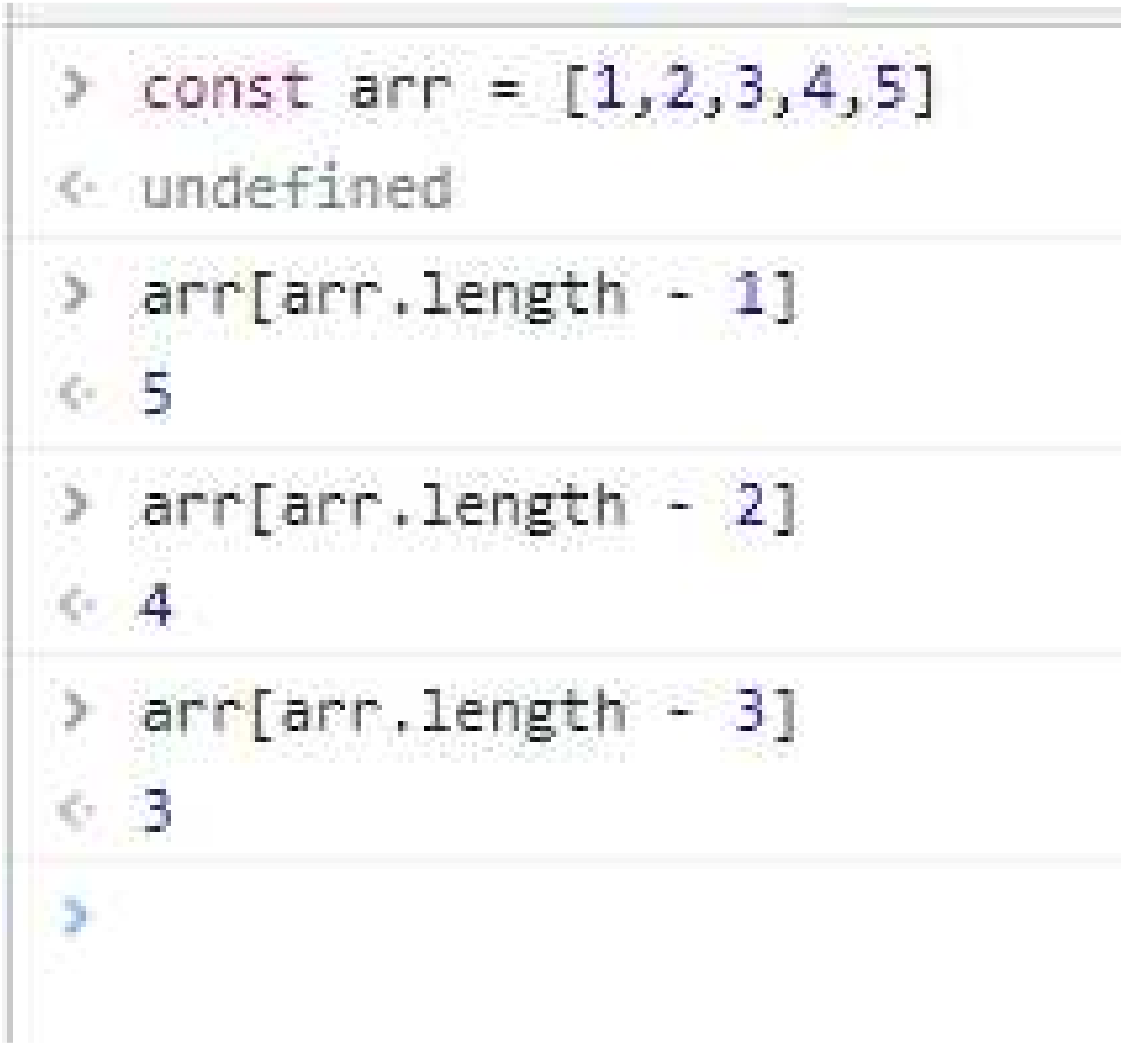


객체 (배열 / 함수 / 그 외 객체)

배열

배열안에 빈값도 요소의 개수에 포함된다

[null, undefined, false, '', Nan]



배열**.unshift(추가할 요소);** : 배열 맨 앞에 값을 추가

배열**.push(추가할 요소);** : 배열 맨 뒤에 값을 추가

*상수더라도 객체 내부의 속성이나 배열의 요소는 바꿀 수 있지만
객체 통째로 바꾸는것은 안된다, 새로운 값을 대입(=) 하지 못한다!

배열**.pop();** : 배열 맨 뒤 값을 제거

배열**.shift();** : 배열 맨 앞 값을 제거

배열**.splice(시작 인덱스,제거 할 요수 개수);** : 시작인덱스에서 부터 제거 할 요소 개수 만큼 제거
배열**.splice(시작 인덱스);** : 시작인덱스에서 부터 마지막 인덱스까지 전부 제거
배열**.splice(시작 인덱스,제거 할 요수 개수,제거 한 자리에 넣을 값);** : 시작인덱스에서 부터 제거 할 요소 개수 만큼 제거 후 그 자리에 다른 값을 넣는다.

배열**.includes(찾는 요소);** : 요소찾기 (true/false)

배열**.indexOf(찾는 요소);** : 앞에서 부터 찾은 해당 요소의 인덱스 위치
배열**.lastIndexOf(찾는 요소);** : 뒤에서 부터 찾은 해당 요소의 인덱스 위치

p127. 1분 퀴즈

반복문에서 '라' 제거

```
> const arr = ['가', '라', '다', '라', '마', '라'];
< undefined
> arr.indexOf('라');
< 1
> arr.splice(1,1);
< ▶ ['라']
> arr.indexOf('라');
< 2
> arr.splice(2,1);
< ▶ ['라']
> arr.indexOf('라');
< 3
> arr.splice(3,1);
< ▶ ['라']
> arr
< ▶ (3) ['가', '다', '마']
>
```

```
> const arr = ['가', '라', '다', '라', '마', '라'];
< undefined
> while(arr.indexOf('라') > -1){
      arr.splice(arr.indexOf('라'), 1);
  }
< ▶ ['라']
> arr
< (3) ['가', '다', '마']
```

함수

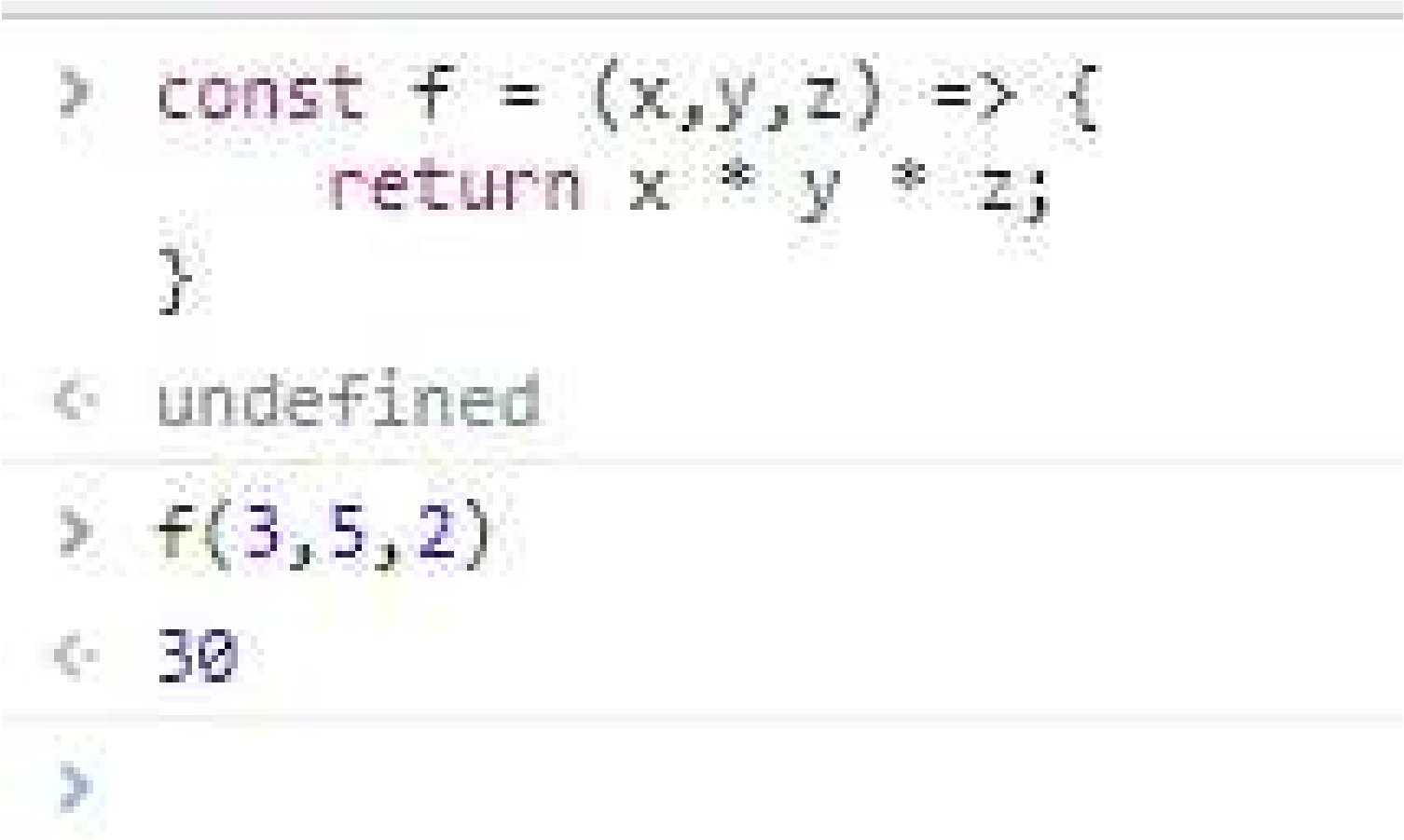
function a**() {}** : 함수 선언문 (함수 이름 a)

const b **= function() {};** : 함수 표현식 (함수 이름 b)

const c **= () => {};** : 화살표 함수 (함수 이름 c)

함수 호출 - argument - **인수**
함수 선언 - parameter - **매개변수**

p135. 1분퀴즈
화살표 함수로 x,y,z를 받아 곱한 값을 반환하는 함수 만들



객체 리터럴

객체 리터럴의 객체 내부에 사용되는 정보 : 속성 (**속성 이름 : 속성 값,**)

접근방법

1. 변수명**.속성이름** : 기본적으로 사용 권**장**
2. 변수**['속성이름']** : 특수한 경우 사용 권장
   '따옴표' 를 빼먹으면 문자열이 아닌 변수가 되므로 전혀 다른 값이 나옴!

속성 수정 : 변수명.속성명 = '수정 값';

속성제거 : **delete** 변수명.속성이름;

메서드 (객체 안에 들어있는 함수)

객체간의 비교

객체끼리는 변수에 넣어놓지 않으면 서로 비교 할 때 모양이 같게 생겼어도 다 다르다.
원시값들은 모양이 같으면 같다.

참조와 복사

객체를 변수에 담으면 **참조**관계가 생긴다!

객체가 아닌 값을 변수에 저장하면 참조관계는 생기지 않는다 = **복사**

p143. 1분퀴즈

'조'값 접근하기