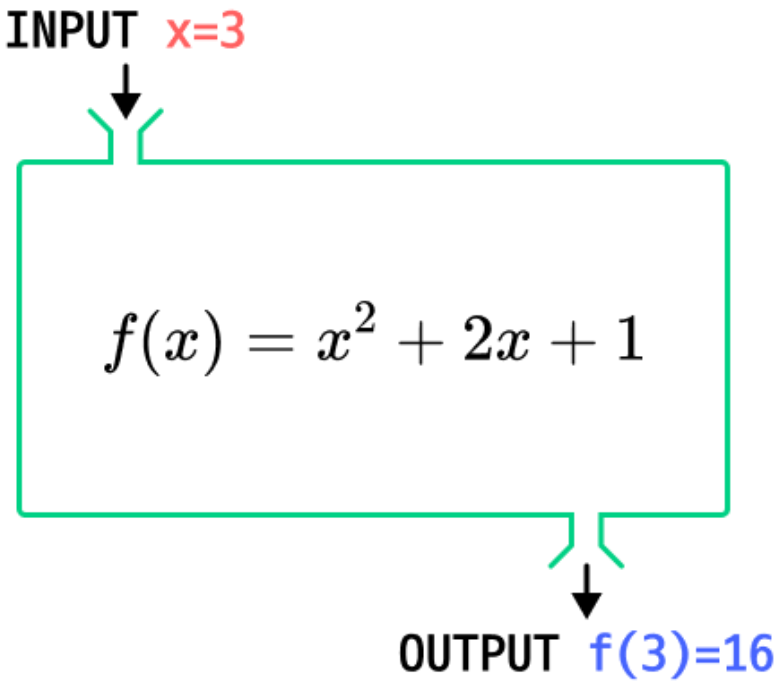
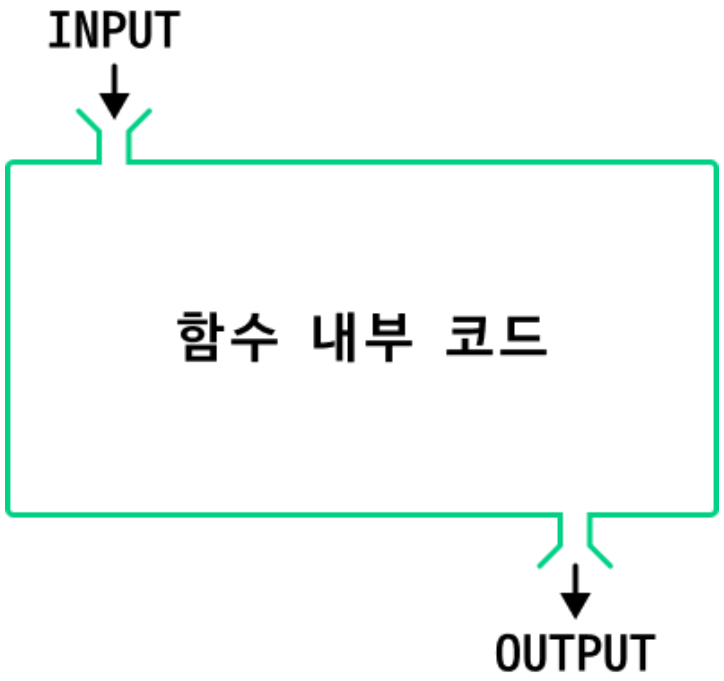
웹 개발자 부트캠프 과정

CodingOn

JavaScript 함수

함수





JavaScript 함수

```
function hello() { }
```

특정 작업을 수행하기 위해 독립적으로 설계된 코드 집합!

JavaScript 함수

```
function hello() { }
```

JavaScript 함수

```
function hello() { }
```

→ **Name (함수 이름)**

: 일반적으로 camelCase 이용

JavaScript 함수

```
function hello() { }
```

➞ **Parameter**

: 함수를 선언할 때 **매개변수(인자)**를 받을 것

JavaScript 함수

```
function hello() { }
```

→ **Body (Block)**

: 함수가 실행되는 Scope 라고도 한다.

함수 선언 방식

명시적 함수 선언

```
function hello() { ... }
```

함수 표현식

```
const hello = function () { ... }
```

실습. 함수 만들기

- [코딩온] HTML, CSS, JS 실습 강의 클릭
  - → [함수 만들기 실습] 커리큘럼 클릭
  - → [실습문제1] 진행

실습. 함수 만들기

- **[코딩온] HTML, CSS, JS 실습** 강의 클릭
  - → **[함수 만들기 실습]** 커리큘럼 클릭
  - → **[실습문제2]** 진행

조건문

조건문

- 특정 조건일 때만 실행하고 싶은 구문이 있을 때 사용합니다.
  - prompt에 입력한 수가 짝수인지 홀수인지 판별하는 프로그램
  - ~ 일 때, ~ 하는 프로그램을 만들 때!

- 조건문 종류
  - if 문
  - 삼항 연산자
  - switch 문

if 문

- 가장 기본적인 조건문
- if문 기본 구조부터 먼저 알아볼까요?

```
if(조건){
  // 괄호 안의 조건이 만족할 때 실행할 문장
}
```

- 조건은 true, false로 결과가 나와요.
  - *a > 10 (변수 a가 10보다 크다면)*
  - *str ===“abcd” (변수 str이 abcd라는 문자열이라면)*
- 조건에 변수 자체를 써도 되는 경우가 있는데, 이 경우는 변수자체가 true 나 false로 판별 가능할 경우입니다.

if와 else

- 기본 if문에서 진화한 형태!

```
if(조건1){
  // 괄호 안의 조건1이 참일 때 실행할 문장
}else{
  //괄호 안의 조건1이 거짓일 때 실행할 문장
}
```

- 소괄호 속의 조건이 참이면 if 이하의 문장, 조건이 거짓이면 else 이하의 문장 실행
- 두개의 문장이 전부 다 실행될 수는 없겠죠?

!! else는 조건을 필요로 하지 않아요!

if와 else if

- 마지막 if문의 모양입니다! else if가 포함된 문장

```
if(조건1){
  // 조건1이 참일 때 실행
}else if(조건2){
  //조건1이 거짓이고, 조건2가 참일 때
}else if(조건3){
  // 조건1 조건2가 거짓이고, 조건3이 참일 때
}else{
  // 조건1 2 3이 모두 거짓일 때 실행
}
```

  - else if도 if처럼 조건이 필요해요.
  - else는 있어도 되고 없어도 됩니다. (else는 조건 x)
  - 하지만 반드시 if문이 가장 먼저 나온 이후에 다른 구문이 나와야 합니다.

if 중첩

- 중첩, if문 안에 또 다른 if문도 들어갈 수 있어요!

```
if ( 조건1 ) {
  if ( 조건2 ) {
    //실행
  } else {
    //실행2
  }
}
```

if 문에서 연산자

- 비교 연산자
  - a == b : a와 b가 동일하면 참
  - a != b : a와 b가 동일하지 않으면 참
  - a < b : a가 b보다 작으면 (b가 a보다 크면) 참
  - a <= b : a가 b보다 작거나 같으면 참

- 논리 연산자
  - a && b : a AND b. a 그리고 b
  - a || b : a OR b. a 또는 b

실습. 연령대별 단어 출력

- **[코딩온] HTML, CSS, JS 실습** 강의 클릭
  - ➡ **[조건문 실습]** 커리큘럼 클릭
  - ➡ **[실습문제1]** 진행

switch 문

- if와 마찬가지로 조건문이지만 switch의 소괄호 안에는 true, false 로 값이 나오는 조건이 아닌 변수 사용
- 변수와 case의 값이 일치한다면 실행

- a가 1일 때, a가 2일 때 각각 실행되는 문장
- break와 default?

```
switch (a) {
  case 1:
    console.log('a는 1입니다.');
    break;
  case 2:
    console.log('a는 2입니다.');
    break;
  default:
    console.log('a값을 모르겠어요.');
    break;
}
```

if 문을 간단하게 만드는 삼항 연산자

- 조건식 ? 조건이 참인 경우 : 조건이 거짓인 경우;
- 한 줄로 간단히 표현 가능!

```
let name = “진형";

// if 문
if (name ==  " 진형") {
  console.log("맞았어요😀 " );
} else {
  console.log( " 틀렸어요 🥵 " );
}

// 3항 연산자
name != “진형 "  ? console.log( " 맞았어요😀") : console.log("틀렸어요🥵");
```

실습. 지금은 오전? 오후?

- **[코딩온] HTML, CSS, JS 실습** 강의 클릭
  - ➡ **[조건문 실습]** 커리큘럼 클릭
  - ➡ **[실습문제2]** 진행

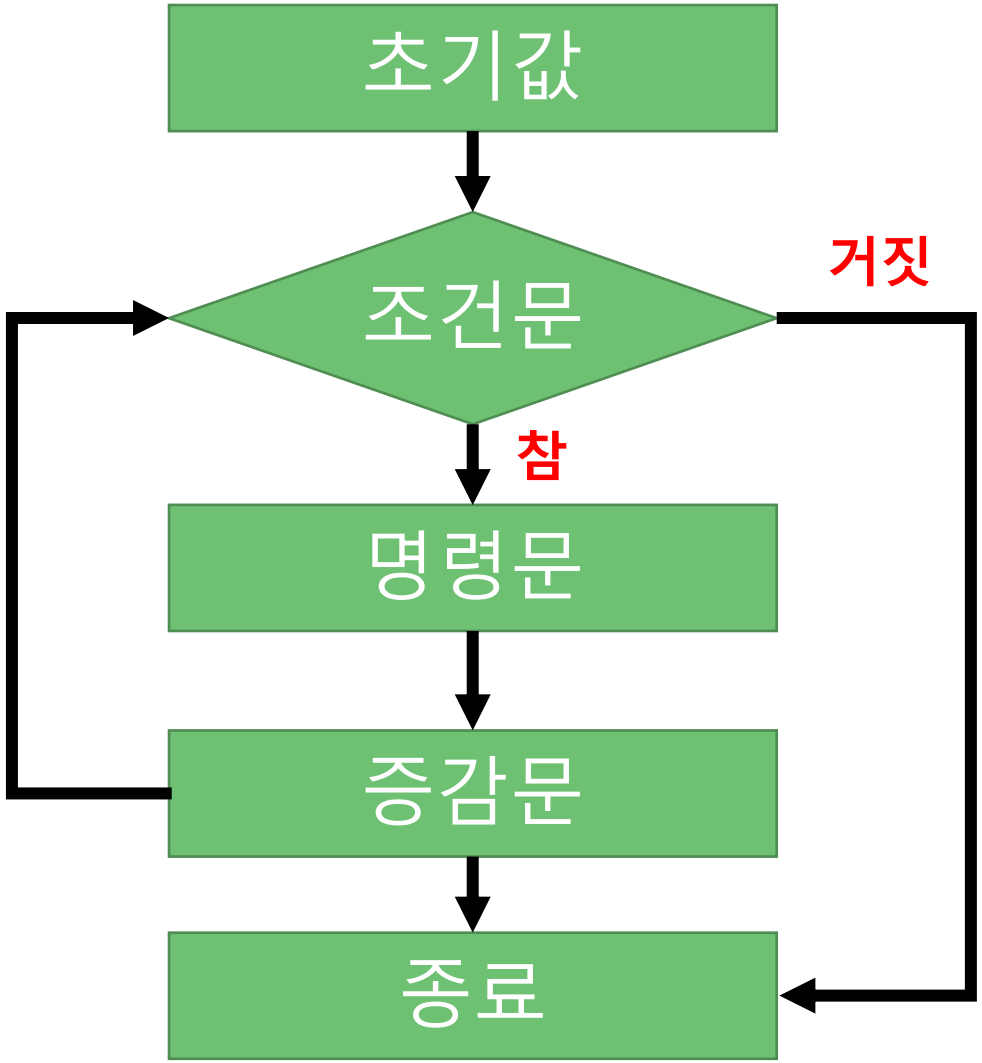
반복문

반복문

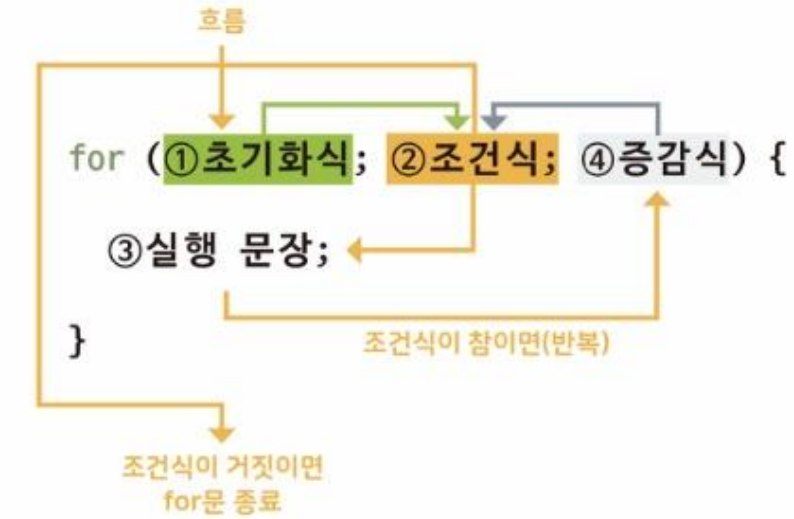
- 특정 코드를 반복하고 싶을 때 사용합니다!
- 100번을 같은 동작을 반복하는데, 100줄을 쓰면 너무나 비효율적이니까요!

- for문
- while문
  - do ~ while …


흐름
for (①초기화식; ②조건식; ④증감식) {
③실행 문장;
}
조건식이 참이면(반복)
조건식이 거짓이면
for문 종료


초기값
조건문
거짓
참
명령문
증감문
종료

for 문




```
// for 문
for(let index = 0; index < 10; index++) {
  console.log("인사를 ", index+1, "번째 드립니다!
😆");
}
```

```
인사를 1 번째 드립니다! 😆
인사를 2 번째 드립니다! 😆
인사를 3 번째 드립니다! 😆
인사를 4 번째 드립니다! 😆
인사를 5 번째 드립니다! 😆
인사를 6 번째 드립니다! 😆
인사를 7 번째 드립니다! 😆
인사를 8 번째 드립니다! 😆
인사를 9 번째 드립니다! 😆
인사를 10 번째 드립니다! 😆
```

중첩 for 문

- 중첩 if 문이 가능하듯 for문도 중첩이 가능하다!

```
for (let i = 0; i < 5; i++) {
  for (let k = 0; k < 3; k++) {
    // 반복할 문장
  }
}
```

while 문

```
while(조건){
  // 조건이 참일 때 실행할 코드
}
```

- for 문과는 달리 값을 제어하는 구문이 기본적으로 포함이 되어 있지 않기 때문에 무한 루프 가능
  - 조건이 항상 참이라면? while문을 빠져나가지 못하고 끝없이 반복하겠지요?
- 따라서 주의하여 사용 필요합니다!

```
// while 문

// 1번 타입, 조건문을 사용
let index = 0;

while (index < 10) {
  console.log("인사를 ", index + 1, "번째 드립니다! 😆");
  index++;
}

// 2번 타입, 조건문을 사용하지 않고 if 문 + break 사용
let index2 = 0;

while (true) {
  console.log("절을 ", index2 + 1, "번째 드립니다! 😆");
  index2++;
  if (index2 == 10) {
    break;
  }
}
```

```
인사를 1 번째 드립니다! 😆
인사를 2 번째 드립니다! 😆
인사를 3 번째 드립니다! 😆
인사를 4 번째 드립니다! 😆
인사를 5 번째 드립니다! 😆
인사를 6 번째 드립니다! 😆
인사를 7 번째 드립니다! 😆
인사를 8 번째 드립니다! 😆
인사를 9 번째 드립니다! 😆
인사를 10 번째 드립니다! 😆
절을 1 번째 드립니다! 😆
절을 2 번째 드립니다! 😆
절을 3 번째 드립니다! 😆
절을 4 번째 드립니다! 😆
절을 5 번째 드립니다! 😆
절을 6 번째 드립니다! 😆
절을 7 번째 드립니다! 😆
절을 8 번째 드립니다! 😆
절을 9 번째 드립니다! 😆
절을 10 번째 드립니다! 😆
```

```
// 구구단 while 버전
let i = 2, j = 1;

while(i < 10) {
    while(j<10) {
        console.log(i, "x", j, "=", i*j);
        j++;
    }
    i++;
    j = 1;
}
```

실습. 구구단 만들기

- [코딩온] HTML, CSS, JS 실습 강의 클릭
  - ➡ [반복문 실습] 커리큘럼 클릭
  - ➡ [실습문제2. 구구단 출력하기] 진행

break & continue

break

- 반복문을 멈추고 빠져나감

```
// break

for(let i = 0; i < 100; i++) {
    if(i==10) {
        console.log("멈춰!");
        break;
    }
    console.log(i);
}
```

```
0
1
2
3
4
5
6
7
8
9
멈춰!
```

continue

- 반복문을 이번 반복 회차를 스킵하고 다음 반복 회차로 진행

```
// continue
let sum = 0;

for(let i = 0; i < 100; i++) {
    if(i%2 == 0) {
        continue;
    }
    sum += i;
}

console.log(sum);
```

```
2500
```

실습. 배수 찾기

- [코딩온] HTML, CSS, JS 실습 강의 클릭
  - ➡ [반복문 실습] 커리큘럼 클릭
  - ➡ [실습문제1. 배수 찾기] 진행

실습. 배수의 합

- [코딩온] HTML, CSS, JS 실습 강의 클릭
  - ➡ [반복문 실습] 커리큘럼 클릭
  - ➡ [실습문제3. 배수의 합] 진행