

K-Digital Training 웹 풀스택 과정

JavaScript (2)

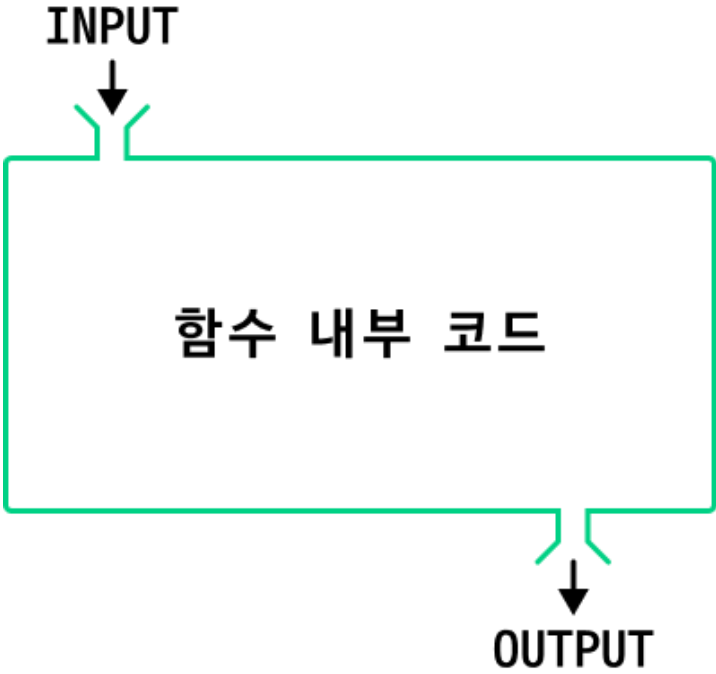
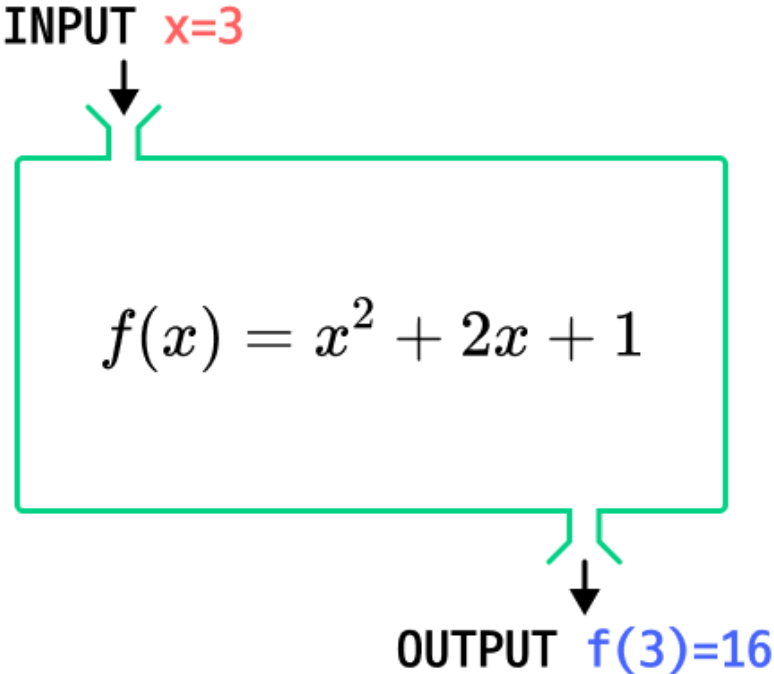
함수

Function!

함수

특정 동작(기능)을 수행하는 일부 코드의 집합(부분)
function

함수



함수 선언문 vs 함수 표현식

```
function sayHello(){  
    console.log('Hello');  
}
```

함수 선언문

함수 선언문 vs 함수 표현식

```
let sayHello = function(){  
  console.log('Hello');  
}
```

함수 표현식

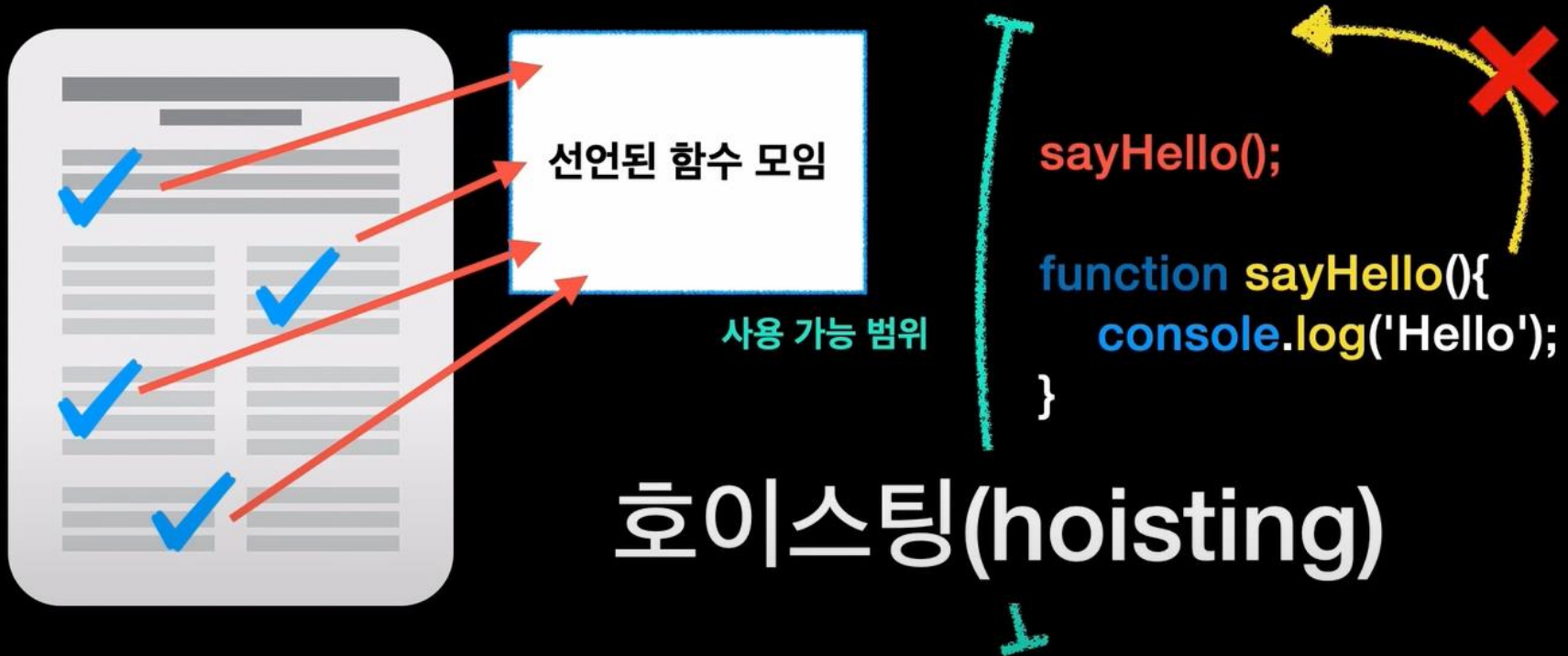
함수 선언문 : 어디서든 호출 가능

sayHello();



```
function sayHello(){  
    console.log('Hello');  
}
```

함수 선언문 : 어디서든 호출 가능



함수 표현식 : 코드에 도달하면 생성

1 ...

2 ...

3 `let sayHello = function(){` 생성
 `console.log('Hello');` 사용가능
 `}`

4 `sayHello();`

화살표 함수(arrow function)

```
let add = function(num1, num2){  
  return num1 + num2;  
}
```

화살표 함수(arrow function)

```
let add = (num1, num2) => {  
  return num1 + num2;  
}
```

화살표 함수(arrow function)

```
let showError = () => {  
    alert('error!');  
}
```

화살표 함수로 만들어 보기

```
// 함수 선언문
function sayHello(name) {
  console.log(`Hello, ${name}`);
}

// 함수 표현식
let sayHello = function (name) {
  console.log(`Hello, ${name}`);
}

// 화살표 함수
let sayHello = (name) => {
  console.log(`Hello, ${name}`);
}
```

실습1. 함수만들기

- multiply() 함수를 만들어주세요!
- 조건
 - 매개변수로 두 개의 숫자를 입력받기
 - 두 인자의 곱을 '반환' 하는 함수를 정의
 - !! 출력이 아닌 반환하는 함수
- 콘솔창에 출력하고 싶다면 아래처럼 테스트 해보기
 - `console.log(multiply(3,7));` // 21
 - `console.log(multiply(2,2));` // 4

실습2. 함수만들기

- square() 함수를 만들어주세요!
- 조건
 - 매개변수로 하나의 숫자를 입력받기
 - 입력받은 수의 제곱을 콘솔창에 출력하는 함수 정의
- 콘솔창에 출력하고 싶다면 아래처럼 테스트 해보기
 - `square(4)` // 16
 - `square(11);` // 121
 - `square(5);` // 25

Onclick!



- 각각의 HTML 요소에 속성 값으로 JS 함수를 연결

```
<body>
  <div class="box"
onclick="test();">click</div>
</body>
```

```
function test() {
  alert("TEST");
}
```

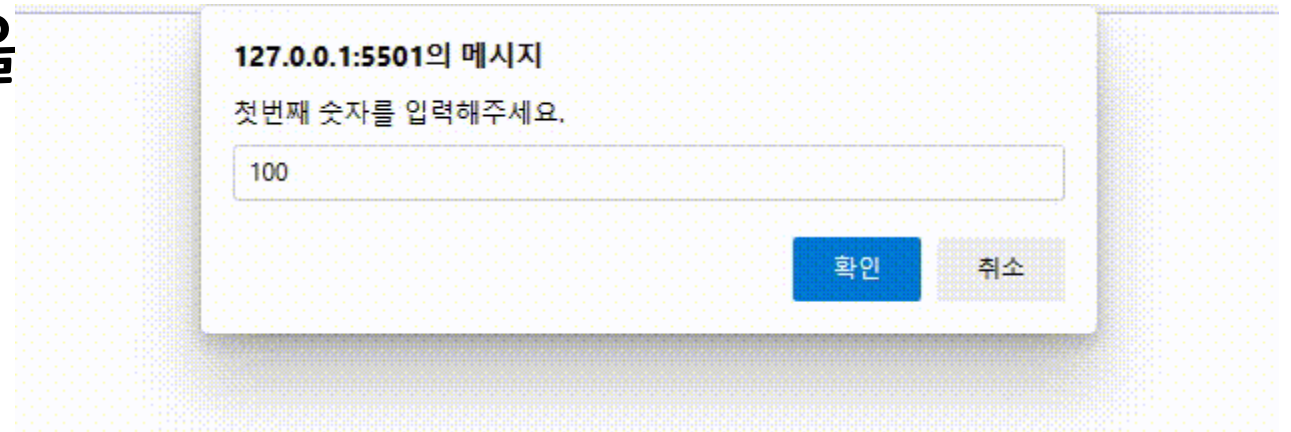
이 페이지 내용:

TEST

확인

실습 3. onclick & 함수만들기

- add, sub, divide, mul 함수 만들기
 - 모든 함수들은 2개의 parameter(a, b)을 받아야 하며, 각각 더하기(a+b), 빼기(a-b), 곱하기(a*b), 나누기(a/b)의 작업을 해야 함.
- 실행할 때 숫자 2개 입력받기 (prompt)
- 버튼 4개를 만들고 각각 버튼을



조건문

Javascript 조건문

특정 조건 만족 시 (조건이 참인 경우) 실행하는 명령의 집합

특정한 조건 속에서 작업을 수행하고 싶을 때 사용

if

switch

조건문

if



IF / ELSE

```
if ( 조건1 ) {  
    // 조건1이 참이라면 실행  
} else {  
    // 조건1이 거짓이라면 실행  
}
```

IF / ELSE IF / ELSE

```
if ( 조건1 ) {  
    // 조건1이 참이라면 실행  
} else if ( 조건2 ) {  
    // 조건2가 참이라면 실행  
} else {  
    // 조건 1과 2가 모두 참이 아닐 때 실행  
}
```

IF 중첩

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

```
let isShow = true;
let checked = false;

if (isShow) {
  console.log('Show!'); // Show!
}

if (checked) {
  console.log('Checked!');
}
```

```
let isShow = true;

if (isShow) {
    console.log('Show!');
} else {
    console.log('Hide?');
}
```

실습4.

if문을 이용해서 console창에 연령대별 단어 출력해보기!

- age 변수 선언, prompt 로 입력받기
 - age가 20 이상: 성인
 - age가 17 이상: 고등학생
 - age가 14 이상: 중학생
 - age가 8 이상: 초등학생
 - age가 0 이상: 유아
-

실습5.

실습 4에 이어 prompt를 이용해 성별 추가로 받기

Age와 성별에 따라 “초등 여학생”, “중등 남학생”, “성인 남자“ 등등 출력하기

조건문

Switch

```
switch ( 변수 ) {  
    case 값1:  
        // 변수와 값1이 일치하면 실행  
        break;  
    case 값2:  
        // 변수와 값2가 일치하면 실행  
        break;  
    default:  
        //일치하는 값이 없을 때 실행  
        break;  
}
```

3항 연산자



IF 문을 간단하게 표현하는 방법

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

```
// 3항 연산자
let name = "길동";

if (name === "길동") {
  console.log("맞았어요😊");
} else {
  console.log("틀렸어요😞");
}

name !== "길동" ? console.log("맞았어요😊") : console.log("틀렸어요😞");
```


실습6.

```
let now = new Date().getHours();
```

위의 코드는 현재 ‘시간’만을 받아오는 코드입니다.

0~ 23까지의 숫자를 반환하고 0이 자정, 12가 정오를 뜻합니다.

now라는 변수에는 현재 시간에 대한 숫자가 저장되어있겠죠?

- 삼항연산자로 지금이 오전인지 오후인지 콘솔창에 출력해보세요!

반복문



Javascript 반복문

똑같은 명령을 일정 횟수만큼 반복해 수행하도록 하는 실행문

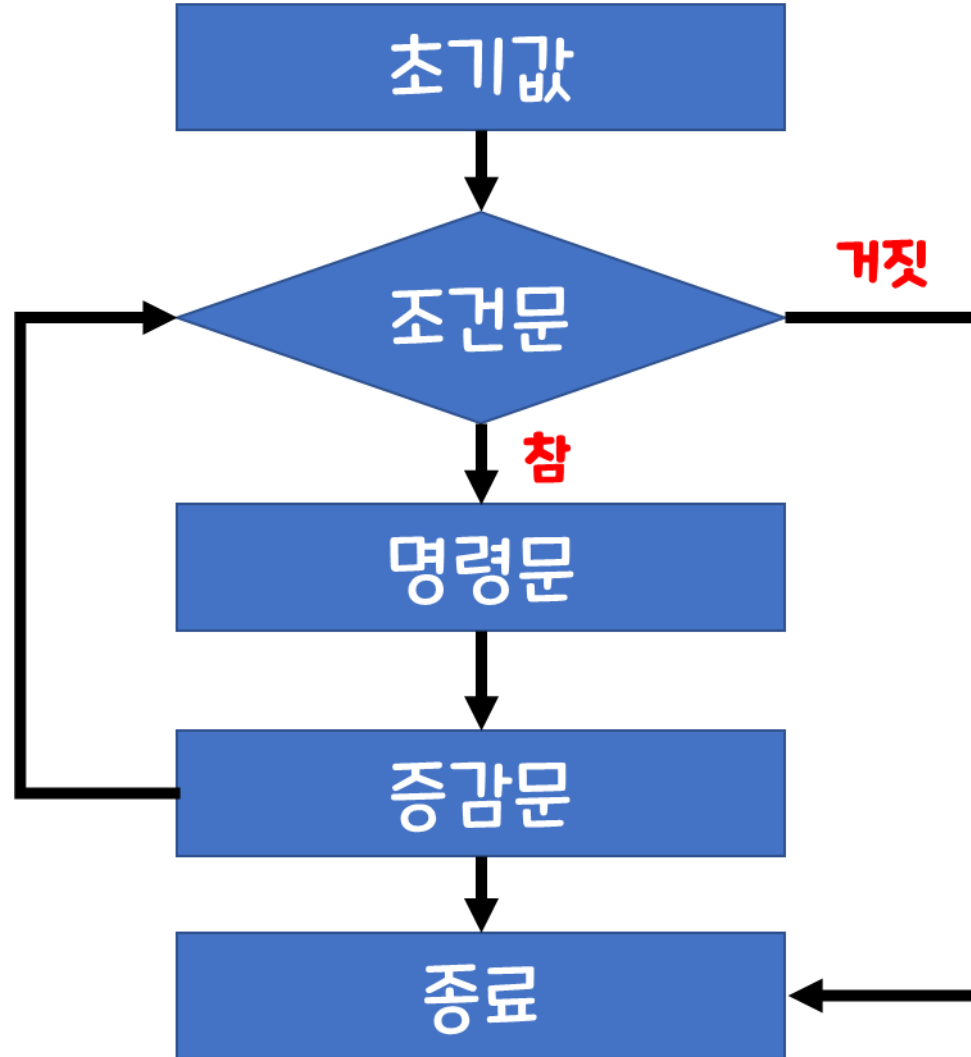
for

while

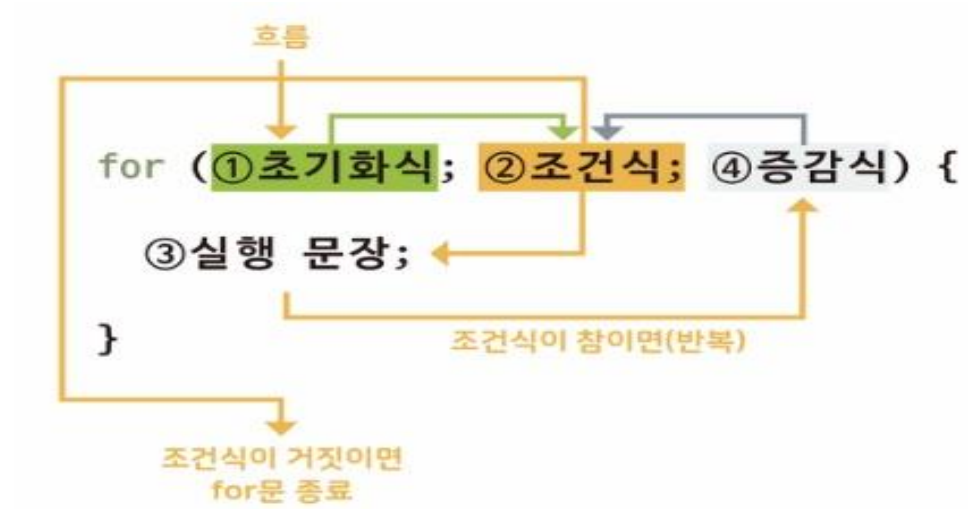
for / in

do / while





For 문



```
// for 문
for(let index = 0; index < 10; index++) {
  console.log("인사를 ", i+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! 😊
```

실습7

- 10000 까지의 숫자 중에서
- 13의 배수면서 홀수인 숫자를 찾아 봅시다!
- + prompt 를 이용해서 입력받은 수 까지 13의 배수면서 홀수인 숫자를 찾는 프로그램을 만들어 봅시다!

- for문을 이용해서 구구단 만들어보기!

---2 단--	index7.js:105
2x 1 = 2	index7.js:107
2x 2 = 4	index7.js:107
2x 3 = 6	index7.js:107
2x 4 = 8	index7.js:107
2x 5 = 10	index7.js:107
2x 6 = 12	index7.js:107
2x 7 = 14	index7.js:107
2x 8 = 16	index7.js:107
2x 9 = 18	index7.js:107
---3 단--	index7.js:105
3x 1 = 3	index7.js:107
3x 2 = 6	index7.js:107
3x 3 = 9	index7.js:107
3x 4 = 12	index7.js:107
3x 5 = 15	index7.js:107
3x 6 = 18	index7.js:107
3x 7 = 21	index7.js:107
3x 8 = 24	index7.js:107
3x 9 = 27	index7.js:107
---4 단--	index7.js:105
4x 1 = 4	index7.js:107
4x 2 = 8	index7.js:107
4x 3 = 12	index7.js:107
4x 4 = 16	index7.js:107
4x 5 = 20	index7.js:107
4x 6 = 24	index7.js:107
4x 7 = 28	index7.js:107
4x 8 = 32	index7.js:107
4x 9 = 36	index7.js:107

---5 단--	index7.js:105
5x 1 = 5	index7.js:107
5x 2 = 10	index7.js:107
5x 3 = 15	index7.js:107
5x 4 = 20	index7.js:107
5x 5 = 25	index7.js:107
5x 6 = 30	index7.js:107
5x 7 = 35	index7.js:107
5x 8 = 40	index7.js:107
5x 9 = 45	index7.js:107
---6 단--	index7.js:105
6x 1 = 6	index7.js:107
6x 2 = 12	index7.js:107
6x 3 = 18	index7.js:107
6x 4 = 24	index7.js:107
6x 5 = 30	index7.js:107
6x 6 = 36	index7.js:107
6x 7 = 42	index7.js:107
6x 8 = 48	index7.js:107
6x 9 = 54	index7.js:107
---7 단--	index7.js:105
7x 1 = 7	index7.js:107
7x 2 = 14	index7.js:107
7x 3 = 21	index7.js:107
7x 4 = 28	index7.js:107
7x 5 = 35	index7.js:107
7x 6 = 42	index7.js:107
7x 7 = 49	index7.js:107
7x 8 = 56	index7.js:107
7x 9 = 63	index7.js:107

---8 단--	index7.js:105
8x 1 = 8	index7.js:107
8x 2 = 16	index7.js:107
8x 3 = 24	index7.js:107
8x 4 = 32	index7.js:107
8x 5 = 40	index7.js:107
8x 6 = 48	index7.js:107
8x 7 = 56	index7.js:107
8x 8 = 64	index7.js:107
8x 9 = 72	index7.js:107
---9 단--	index7.js:105
9x 1 = 9	index7.js:107
9x 2 = 18	index7.js:107
9x 3 = 27	index7.js:107
9x 4 = 36	index7.js:107
9x 5 = 45	index7.js:107
9x 6 = 54	index7.js:107
9x 7 = 63	index7.js:107
9x 8 = 72	index7.js:107
9x 9 = 81	index7.js:107

반복문

while

while 문

- while(조건문) {

실행할 코드(명령문)

}

- For 문과는 달리 조건을 변경하는 구문이 기본적으로 포함되어 있지 않기 때문에 무한 루프 가능
 - 주의하여 사용 필요
-

```
// 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!	😊

while

```
let i = 0;
```

```
while (i < 10) {
```

```
// 코드
```

```
}
```

while

```
let i = 0;
```

```
while (i < 10) {  
  // 코드  
  i++;  
}
```

```
// 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;
}
```

반복문

제어

break, continue

break

: 멈추고 빠져나옴

continue

: 멈추고 다음 반복으로 진행

break

- 반복문을 멈추고 밖으로 빠져 나감

```
// 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 한 번만 멈추고 다음으로 진행

```
// continue
let sum = 0;

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

console.log(sum);
```

2500

실습9

정수 n 이 있을 때 2또는 3의 배수의 총합을 구하시오.
(단, n 은 $0 < n < 100$)