웹 앱 개발을 위한 Javascript 기초

4. 자바스크립트 함수

한국기술교육대학교
온라인평생교육원

학습내용

1. 함수 기본
2. 함수 활용

학습목표

1. 함수의 기본 형태, 함수의 매개 변수와 반환에 대해 설명할 수 있다.
2. 함수를 변수에 저장할 수 있다는 점, 콜백 함수 사용 방법, 자바스크립트의 표준 내장 함수에 대해 설명할 수 있다.

1. 함수 기본

함수 생성 방법

익명 함수

- 이름을 붙이지 않고 함수 생성
- 함수를 호출하면 함수 내부의 코드 덩어리가 모두 실행

```
let <함수 이름> = function ( ) { };

let 함수 = function ( ) {
    console.log("함수의 첫 번째 줄");
    console.log("함수의 두 번째 줄");
);

함수();
Console.log(함수);
```

함수를 생성합니다.

함수를 호출합니다.

함수 자체를 출력합니다.

선언적 함수

- 이름을 붙여 함수를 생성

```
let <함수 이름> = function ( ) { };

let 함수 = function ( ) {
    console.log("함수의 첫 번째 줄");
    console.log("함수의 두 번째 줄");
);

함수();
Console.log(함수);
```

함수를 생성합니다.

함수를 호출합니다.

함수 자체를 출력합니다.

1. 함수 기본

함수 생성 방법

```
function 함수 ( ) {
    console.log("함수의 첫번째 줄");
    console.log("함수의 두번째 줄");
};
함수();
console.log(함수);
```

```
함수의 첫번째 줄
함수의 두번째 줄
```

```
ƒ 함수 ( ) {
    console.log("함수의 첫번째 줄");
    console.log("함수의 두번째 줄");
}
```

```
undefined
```

- 화살표 함수
  - '하나의 표현식을 리턴하는 함수'를 만들 때는 중괄호 생략 가능
  - 익명 함수 예제를 화살표 함수로 바꾸기

```
() => { }

let 함수 = () => {
    console.log("함수의 첫 번째 줄");
    console.log("함수의 두 번째 줄");
);

함수();
Console.log(함수);
```

함수를 생성합니다.

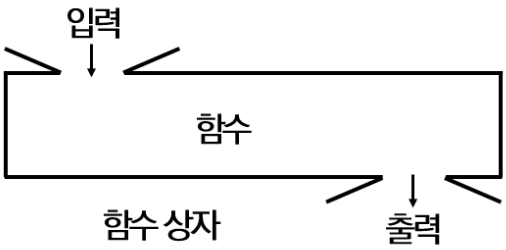
함수를 호출합니다.

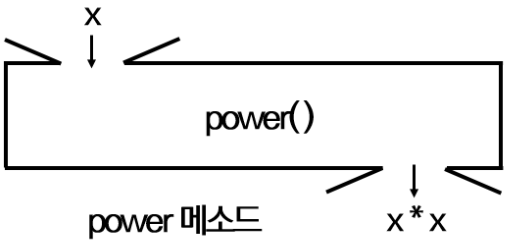
1. 함수 기본

함수의 기본 형태

* 기본 형태

```
function 〈함수 이름〉(〈매개 변수〉) {
    〈함수 코드〉
    return 〈리턴 값〉
}
```





함수의 기본 형태

* 매개 변수로 넣은 숫자를 제곱하는 power( ) 함수 생성

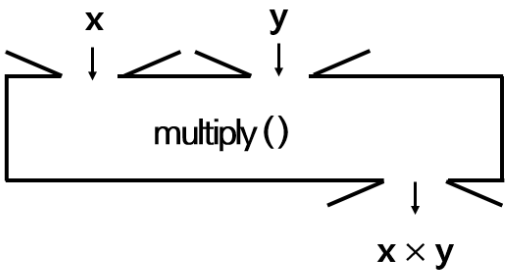
```
function power(x) {
    return x * x;
}
console.log(power(10));
console.log(power(20));
```

```
100
400
undefined
```

1. 함수 기본

함수의 기본 형태

* 매개 변수가 여러 개인 함수

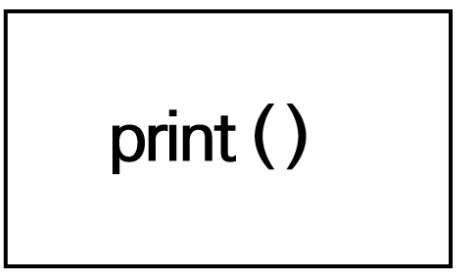



```
function multiply(x,y) {
      return x × y;
}

console.log(multiply(52, 273));
console.log(multiply(103, 32));
```

함수의 기본 형태

* 리턴 없는 함수




```
funtion print(message) {
    console.log(`*${message}"(이)라고 말했습니다!`);
}

print(*안녕하세요*);
Print(*뿌잉뿌잉*);
```

2. 함수 활용

함수의 기본 활용 형태

- 리턴하는 함수의 기본 형태
- 매개 변수와 리턴(1) – min부터 max까지 숫자를 더해 리턴

```
function 함수이름 (〈매개 변수〉, 〈매개 변수〉) {
    let output = 〈초깃값〉 ;
    // output 계산
    return output;
}
```

```
function sum(min, max) {
    let output = 0;
    for (let i = min; i <= max; i++) {
        output += i;
    }
    return output;
}
console.log(sum(1, 100));
```

```
5050
undefined
```

함수의 기본 활용 형태

- 리턴하는 함수의 기본 형태
- min부터 max까지 숫자를 곱해 리턴하는 함수를 생성 호출

```
function 함수이름 (〈매개 변수〉, 〈매개 변수〉) {
    let output = 〈초깃값〉 ;
    // output 계산
    return output;
}
```

```
function multiply(min, max) {
    let output =1;
    for (let i = min; i <= max; i++) {
        output *= i;
    }
    return output;
}
console.log(multiply(1,10));
```

```
3628800
undefined
```

2. 함수 활용

함수 매개 변수 초기화

* 매개 변수를 입력하지 않고 함수 호출
* 실행하면 undefined가 출력

```
// 함수를 선언합니다.
function print(name, count) {
      console.log(‘${name}이/가 ${count}개
있습니다.‘)
}

// 함수를 호출합니다.
print(“사과”, 10);
Print(“사과”);
```

함수 매개 변수 초기화

* 조건문을 활용한 매개 변수 초기화
* 조건문으로 매개 변수를 확인, count가 undefined일 때 1로 초기화

```
//함수를 선언합니다.
function print(name, count){
    if(!count) {
        count = 1;
    }
    //함수 본문
    console.log(`${name}이/가 ${count}개 있습니다.`)
}
//함수를 호출합니다.
print("사과");
```

```
사과이/가 1개 있습니다.
undefined
```

2. 함수 활용

함수 매개 변수 초기화

* 짧은 초기화 조건문을 활용한 매개 변수 초기화
* 짧은 초기화 조건문으로 매개 변수를 확인, count가 undefined일 때 1로 초기화

```
//함수를 선언합니다.
function print(name, count){
    count = count || 1;
    //함수 본문
    console.log(`${name}이/가 ${count}개 있습니다.`)
}
//함수를 호출합니다.
print("사과");
```

```
사과이/가 1개 있습니다.
undefined
```

콜백 함수

* 함수의 매개 변수로 전달되는 함수
* 콜백 함수 사용

```
> //함수를 선언합니다.
  function callTenTimes(callback) {
      // 10회 반복합니다.
      for (let i = 0; i < 10; i++) {
          //매개 변수로 전달된 함수를 호출합니다.
          callback();
      }
  }
  //변수를 선언합니다.
  callTenTimes(function () {
      console.log('함수 호출');
  });
```

```
10 함수 호출
undefined
```

2. 함수 활용

표준 내장 함수

- 자바스크립트에서 기본적으로 지원하는 함수
- 숫자 변환 함수

함수	설명
parseInt( )	문자열을 정수로 변환합니다.
parseFloat( )	문자열을 실수로 변환합니다.

- parseInt( ) 함수와 parseFloat( ) 함수
- 문자열을 숫자로 변환

```
//변수를 선언합니다.
let inputA = "52";
let inputB = "52.273";
let inputC = "1401동"
//parseInt() 함수의 기본적인 사용
console.log(parseInt(inputA))
//parseInt() 함수dhk parseFloat() 함수의 차이
console.log(parseInt(inputB))
console.log(parseFloat(inputB))
//문자열 뒤에 숫자가 아닌 문자가 포함되어 있을 때
console.log(parseInt(inputC));
```

```
52
52
52.273
1401
undefined
```

- 타이머 함수
  - '특정 시간 후에' 또는 '특정 시간마다' 어떤 일을 할 때 사용
  - 시간은 밀리초로 지정. 1초를 나타내려면 1000(밀리초)을 입력

함수	설명
setTimeout(함수, 시간)	특정 시간 후에 함수를 실행합니다.
setInterval(함수, 시간)	특정 시간마다 함수를 실행합니다.

2. 함수 활용

표준 내장 함수

* 타이머 함수

- 1초 후에 '1초가 지났습니다.', 1초마다 '1초 마다 호출됩니다.'를 출력

```
> //1초 후에
  setTimeout(function () {
      console.log("1초가 지났습니다.");
  }, 1000);
  //1초마다
  setInterval(function () {
      console.log("1초 마다 호출됩니다.");
  }, 1000);
< 9
  1초가 지났습니다.
  1초 마다 호출됩니다.
```

종료 : Ctrl + C
실행 화면에서 ^C는 Ctrl + C 를 눌러서 종료해야 함
'1초 마다 호출됩니다.' 가 반복 실행됨
웹브라우저 개발자도구의 경우 브라우저를 종료

함수	설명
clearInterval(아이디)	특정 시간마다 실행하던 함수 호출을 정지합니다.

표준 내장 함수

* 타이머 함수

```
> // 1초마다
  let id = setInterval(function () {
      console.log("출력합니다.");
  }, 1000);
  // 3초 후에
  setTimeout(function () {
      //타이머를 제거합니다.
      clearInterval(id);
  }, 3000);
< 4
  출력합니다.
```

2. 함수 활용

익명 함수와 선언적 함수의 생성 순서

* 변수 덮어쓰기

```
let 변수;
변수 = 10;
변수 = 20;

console.log(변수)
```

20이 변수에 저장

익명 함수와 선언적 함수의 생성 순서

* 함수 덮어쓰기

```
let 변수;
변수 = function ( ) { console.log(*첫 번째 함수*);};
변수 = function ( ) { console.log(*두 번째 함수*);};

함수( );
```

두 번째 함수가 실행

정리하기

1. 함수 기본

- 함수의 생성 방법

- 익명 함수 : 이름을 붙이지 않고 함수 생성
- 선언적 함수 : 이름을 붙여 함수를 생성
- 화살표 함수 : '하나의 표현식을 리턴하는 함수'를 만들 때는 중괄호 생략 가능

- 함수의 기본 형태

2. 함수 활용

- 기본 형태, 매개 변수로 넣은 숫자를 제곱하는 power( ) 함수 생성, 매개 변수가 여러 개인 함수, 리턴 없는 함수, 리턴하는 함수의 기본 형태, 매개 변수를 입력하지 않고 함수 호출, 조건문을 활용한 매개 변수 초기화, 콜백 함수(함수의 매개 변수로 전달되는 함수, 표준 내장 함수(자바스크립트에서 기본적으로 지원하는 함수)로 활용됨