

영인터내셔널 사내교육 (웹 프로그래밍)

기본

자바스크립트

Contents

학습목표

- 자바스크립트에서 사용하는 기본 용어를 이해합니다.
- 기본 출력 방법을 익힙니다.
- 기본 자료형, 변수, 상수를 배우고 관련된 연산자의 사용 방법을 익힙니다.
- 자료형 변환 방법을 익힙니다.

Contents

내용

- 기본 용어
- 출력
- 기본 자료형
- 변수
- 복합 대입 연산자
- 증감 연산자
- 자료형 검사
- undefined 자료형
- 강제 자료형 변환
- 자동 자료형 변환
- 일치 연산자
- 상수

1. 기본 용어

■ 표현식과 문장

■ 표현식

```
273
```

```
10 + 20 + 30 * 2
```

```
"JavaScript Programming"
```

- 문장 : 표현식이 하나 이상 모일 경우, 마지막에 종결 의미로 세미콜론(;)
- 프로그램 : 문장이 모이면 프로그램이 됨

1. 기본 용어

■ 식별자

- 이름을 붙일 때 사용하는 단어, 변수와 함수 이름 등으로 사용
 - 키워드를 사용 안됨
 - 특수 문자는 _와 \$만 허용
 - 숫자로 시작하면 안됨
 - 공백은 입력하면 안됨

```
alpha  
alpha10  
_alpha  
$alpha  
ALPha  
ALPHA
```

X

```
break  
273alpha  
has space
```

O

1. 기본 용어

■ 식별자

■ 식별자 사용 규칙

- 생성자 함수의 이름은 항상 대문자로 시작
- 변수, 함수, 속성, 메소드의 이름은 항상 소문자로 시작
- 여러 단어로 된 식별자는 각 단어의 첫 글자를 대문자로 함

```
will out      // willOut  
will return   // willReturn  
i am a boy    // iAmABoy
```

1. 기본 용어

■ 주석

- 프로그램의 진행에 영향을 주지 않는 코드

방법	표현
한 줄 주석 처리	// 주석
여러 줄 주석 처리	/* 주석 주석 */

```
// 주석은 코드의 실행에 영향을 주지 않습니다.  
/*  
console.log("JavaScript Programming")  
console.log("JavaScript Programming")  
console.log("JavaScript Programming")  
*/
```

2. 출력

■ 출력 메소드

- console 객체의 log() 메소드 사용 : console.log() 메소드

```
console.log("문자열")
```

- 자바스크립트의 기본 출력
 - 문자열 'JavaScript Programming' 출력하기

```
console.log("JavaScript Programming");
```

실행 결과

JavaScript Programming

2. 출력

■ 출력

- 곧바로 문장을 입력해서 출력

> "안녕" + "하세요"

'안녕하세요'

> 52 + 273

325

>

> 문장

출력_결과

3. 기본 자료형

■ 숫자

■ 가장 기본적인 자료형

```
console.log(52);  
console.log(52.271);
```

연산자	설명
+	덧셈 연산자
-	뺄셈 연산자
*	곱셈 연산자
/	나눗셈 연산자

```
> 52 + 273  
325
```

3. 기본 자료형

■ 연산자 우선순위

```
console.log(5 + 3 * 2);
```

■ 나머지 연산자

연산자	설명
%	나머지 연산자

```
> 10 % 5  
0  
> 7 % 3  
1
```

3. 기본 자료형

■ 문자열

- 문자의 집합
- 문자열 생성시 큰따옴표나 작은따옴표를 사용

```
> "안녕하세요"  
'안녕하세요'  
> '안녕하세요'  
'안녕하세요'
```

```
> console.log("This is 'String'")  
This is 'String'  
undefined  
> console.log('This is "String"')  
This is "String"  
undefined
```

3. 기본 자료형

- 이스케이프 문자
 - 따옴표를 문자 그대로 사용 가능
 - 문자열 줄바꿈 할 경우 사용

```
> console.log("This is \"String\"")
This is "String"
undefined
> console.log('This is \'String\'')
This is 'String'
undefined
```

```
> console.log("동해물과 백두산이\n마르고 닳도록")
동해물과 백두산이
마르고 닳도록
undefined
```

3. 기본 자료형

■ 문자열 합하기

연산자	설명
+	문자열 연결 연산자

■ 문자열 연결 연산자

```
console.log("가나다" + "라마" + "바사아" + "자차카타" + "파하");
```

실행 결과

가나다라마바사아자차카타파하

3. 기본 자료형

- 문자 선택 연산자

연산자	설명
문자열[숫자]	문자 선택 연산자

- 문자 선택 연산자
- '안녕하세요' 문자열의 0번째, 1번째, 3번째에 있는 문자를 선택

```
console.log("안녕하세요"[0]);  
console.log("안녕하세요"[1]);  
console.log("안녕하세요"[3]);
```

실행 결과

안
녕
세

3. 기본 자료형

- 템플릿 문자열

```
> `안녕하세요`  
'안녕하세요'
```

```
> `52 + 273 = ${52 + 273}`  
'52 + 273 = 325'
```

```
> `올해는 ${new Date().getFullYear()}년입니다.`  
'올해는 2022년입니다.'
```


3. 기본 자료형

■ 불

- 참과 거짓의 표현 : true와 false

```
> true  
true  
> false  
false
```

```
52 < 273  
52 > 273
```

- 비교 연산자

연산자	설명
==	같습니다.
!=	다릅니다.
>	왼쪽 피연산자가 큼니다.
<	오른쪽 피연산자가 큼니다.
>=	왼쪽 피연산자가 크거나 같습니다.
<=	오른쪽 피연산자가 크거나 같습니다.

3. 기본 자료형

■ 불과 비교 연산자

```
console.log(52 < 273);  
console.log(52 > 273);  
console.log("하마" < "가방");
```

실행 결과

```
true  
false  
false
```

연산자	설명
!	논리 부정 연산자
	논리합 연산자
&&	논리곱 연산자

3. 기본 자료형

- 논리 부정 연산자

```
console.log(52 < 273);  
console.log(52 > 273);  
console.log("하마" < "가방");
```

실행 결과

```
true  
false  
false
```

3. 기본 자료형

- 논리합 연산자 (이항 연산자)

왼쪽 피연산자	오른쪽 피연산자	결과
true	true	true
true	false	true
false	true	true
false	false	false

- 논리곱 연산자 (이항 연산자)

왼쪽 피연산자	오른쪽 피연산자	결과
true	true	true
true	false	false
false	true	false
false	false	false

4. 변수

- 변수 : 값을 저장할 때 사용하는 식별자, 변수 선언 후 변수에 값을 할당
 - 변수 선언
 1. 변수를 선언합니다.
 2. 변수에 값을 할당합니다.
 - 변수 pi를 선언

```
> let pi;  
undefined
```

- 변수 pi에 값을 할당

```
> let pi;  
undefined  
> pi = 3.14159265;  
undefined
```

4. 변수

- 변수 초기화

```
> let pi = 3.14159265;  
undefined
```

- 변수 활용

```
> let pi = 3.14159265;  
undefined  
> console.log(pi);  
3.14159265  
undefined
```

5. 복합 대입 연산자

- 변수에 사용할 수 있는 몇 개의 특별한 연산자가 존재
 - $a += 10$ 는 $a = a + 10$ 과 결과가 같음

연산자	설명
<code>+=</code>	숫자 덧셈 후 대입 연산자
<code>-=</code>	숫자 뺄셈 후 대입 연산자
<code>*=</code>	숫자 곱셈 후 대입 연산자
<code>/=</code>	숫자 나눗셈 후 대입 연산자

연산자	설명
<code>+=</code>	문자열 연결 후 대입 연산자

5. 복합 대입 연산자

■ 숫자와 관련된 복합 대입 연산자

```
let output = 0;  
output += 52;  
output += 273;  
output += 103;  
  
console.log(output);
```

실행 결과

428

```
let output = 0;  
output = output + 52;  
output = output + 273;  
output = output + 103;  
  
console.log(output);
```


6. 증감 연산자

- 변수 number를 초기화하고, ++ 연산자와 -- 연산자를 사용
- 각 연산자에서 변수 값이 1만큼 변경됨

연산자	설명
변수++	기존 변수 값에 1을 더합니다(후위).
++변수	기존 변수 값에 1을 더합니다(전위).
변수--	기존 변수 값에서 1을 뺍니다(후위).
--변수	기존 변수 값에서 1을 뺍니다(전위).

```
let number = 10;
```

```
number++;  
console.log(number);  
number--;  
console.log(number);
```

실행 결과

11
10

7. 자료형 검사

■ 자료형 확인 연산자

연산자	설명
typeof	해당 변수의 자료형을 추출합니다.

```
> typeof 10
'number'
> typeof "문자열"
'string'
```

■ 보통 연산자 뒤에 괄호를 붙임

```
> typeof(10)
'number'
> typeof("문자열")
'string'
```

7. 자료형 검사

■ 자바스크립트의 여섯 가지 자료형

```
> // 문자열
> typeof('String')
'string'

> // 숫자
> typeof(273)
'number'

> // 불
> typeof(true)
'boolean'

> // 함수
> typeof(function () { })
'function'

> // 객체
> typeof({})
'object'

> // undefined
> let alpha
> typeof(alpha)
'undefined'
```

8. undefined 자료형

■ undefined 자료형

- 변수를 선언했으나 초기화하지 않은 자료형

```
> let a  
undefined  
> typeof(a)  
"undefined"  
  
> typeof(b)  
"undefined"
```

초기화하지 않은 변수입니다.

선언하지 않았던 식별자입니다.

9. 강제 자료형 변환

■ 강제 자료형 변환

■ 강제 자료형 변환 함수

함수	설명
Number()	숫자로 자료형 변환합니다.
String()	문자열로 자료형 변환합니다.
Boolean()	불로 자료형 변환합니다.

■ String() 함수

```
> String(52)
"52"
> String(273)
"273"

> String(true)
"true"
> String(false)
"false"
```

숫자는 그대로 문자열로 바뀝니다.

불도 그대로 문자열로 바뀝니다.

10. 자동 자료형 변환

■ 숫자와 문자열 자료형 자동 변환

- 숫자와 문자열에 '+' 연산자를 적용하면 자동으로 숫자가 문자열로 변환
- 숫자와 문자열 자료형 변환 – 덧셈 연산자

```
console.log(52 + 273);  
console.log("52" + 273);  
console.log(52 + "273");  
console.log("52" + "273");
```

실행 결과

```
325  
52273  
52273  
52273
```

11. 일치 연산자

■ 일치 연산자

■ 자료형까지 검사

연산자	설명
===	자료형과 값이 같은지 비교합니다.
!==	자료형과 값이 다른지 비교합니다.

■ 비교 연산자와 일치 연산자의 차이

```
console.log(`52 == "52": ${52 == "52"}`);  
console.log(`52 === "52": ${52 === "52"}`);  
console.log();  
console.log(`0 == "" : ${0 == ""}`);  
console.log(`0 === "" : ${0 === ""}`);
```

실행 결과

```
52 == "52": true  
52 === "52": false  
  
0 == "": true  
0 === "": false
```

12. 상수

■ 상수

- 상수 : '항상 같은 수'라는 의미, 변수와 반대되는 개념
- `const` : 상수(constant)를 만드는 키워드
- 변하지 않을 대상에 상수를 적용