



Chapter 02

자바스크립트 기초 문법-2



변수

변수의 선언과 초기화

■ 변수의 선언과 초기화

- 자바스크립트는 변수 선언 시 자료형을 지정하지 않음
- var라는 키워드를 사용해 선언할 수 있음
 - 초기화하지 않고 변수만을 사용할 경우 반드시 var 키워드를 사용해야 함
 - 선언과 동시에 초기화할 경우 var 키워드를 사용하지 않아도 됨
- 변수는 선언과 동시에 초기화 할 수 있음
 - 변수의 선언 후 초기화

```
var 변수명;           // 값을 가지지 않는 변수의 선언  
변수명 = 값;          // 변수 선언 후 초기화
```

```
var box;  
box = 100;
```

- 변수의 선언과 동시에 초기화

```
[var] 변수명 = 값;    // 값을 가지지 않는 변수의 선언
```

```
box = 100; 또는  
var box = 100;
```

변수가 가질 수 있는 자료형

■ 문자형 데이터

- 한글, 한자, 영문자, HTML string 등의 데이터
 - 반드시 이중 인용부호 내에 표현해야 함
 - 숫자형 자료를 이중 인용부호 내에 표현할 경우 문자열로 인식하므로 연산에 참여 불가능

```
[var] 변수명 = "문자형 데이터";
```

```
var s="javascript";  
var num="100";  
var tag="<h1> String </h1>";
```

변수가 가질 수 있는 자료형

■ 숫자형 데이터

- 정수나 실수 값을 가지며 10진수, 8진수, 16진수 표현이 가능

[var] 변수명 = 숫자형 데이터;

```
var num1 = 10 ;           // 숫자형 데이터 10을 10진수로 표현
var num1 = 012 ;          // 숫자형 데이터 10을 8진수로 표현
var num1 = 0Xa ;          // 숫자형 데이터 10을 16진수로 표현
```

- 인용부호 내의 숫자형 데이터(문자형 데이터)를 실제 숫자형 데이터로 지정하는 방법
 - Number() 메서드나 parseInt() 메서드를 사용

```
var num1 = "10";           // 문자형 데이터 10
var num2 = Number(num1);    // 문자형 데이터 10을 숫자형 데이터로 전환 (= parseInt(num1))
document.write(num2+10);    // 20이 출력됨
```

변수가 가질 수 있는 자료형

■ 논리형 데이터

- 값으로 참이나 거짓을 가지는 데이터

- C와 C++에서와는 달리 1과 0을 가지지 않고 `true`와 `false`를 가짐

[var] 변수명 = `true/false` 또는
`Boolean(데이터)`

```
var s=true;           // 논리형 변수 s에 true가 저장됨  
var t=10>=100;        // (10>=100)이 거짓이므로 논리형 변수 t에 false가 저장됨
```

- `Boolean()` 메서드 내의 0, `null`, `undefined`를 제외한 모든 데이터가 올 경우 `true`를 반환

```
var s = Boolean(0);    // false를 반환하므로 변수 s에 false가 저장됨  
var t = Boolean("홍길동"); // true를 반환하므로 변수 t에 true가 저장됨
```

변수가 가질 수 있는 자료형

• 논리형 자료형의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <script type="text/javascript">
    var a = true;
    var b = false;
    var c = 10 > 5;
    var d = Boolean(null);

    document.write(a, "<br>");
    document.write(b, "<br>");
    document.write(c, "<br>");
    document.write(d, "<br>");
  </script>

</head>
<body>

</body>
</html>
```

2-boolean.htm (p. 40)

```
true
false
true
false
```

변수가 가질 수 있는 자료형

■ null 자료형과 undefined 자료형

• undefined 자료형

- var 키워드를 사용해 변수는 선언되어 있지만 아무런 값도 지정되지 않은 경우
- 차후에 변수가 어떤 자료형의 데이터가 저장될 지 몰라 변수를 선언하기만 할 경우 사용

```
var s;
```

• null 자료형

- 변수의 값으로 null이 지정된 경우
- 주로 변수에 저장된 데이터를 비우고자 할 때 사용

```
var t = null;
```


변수가 가질 수 있는 자료형

■ typeof

- 특정 데이터 또는 변수의 자료형을 출력하기 위해 사용

typeof 변수 (또는 데이터)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <script type="text/javascript">
    var num = 100;
    var str = "자바스크립트";
    document.write(typeof num, "<br>");
    document.write(typeof str);
  </script>

</head>
<body>
</body>
</html>
```

2-typeof.htm (p. 41)

number
string

변수 선언 시 유의 사항

■ 변수 선언 시 유의사항

- 변수명의 첫 글자는 \$, _(언더바), 영문자만 가능함

```
var 2time = 0;    (X)  
var $time = 0;    (O)
```

- 변수명의 첫 글자 다음 문자들은 영문자, 숫자, \$, _(언더바)가 아니어도 됨

```
var 100num = 100; (X)  
var num100 = 100; (O)
```

- 변수명으로 예약어를 사용할 수 없음

```
var document = 100; (X)  
var num = 100;      (O)
```

- 변수명은 대소문자를 구분함

```
var num = 100; document.write(Num)    (X)  
var num = 100; document.write(num)    (O)
```

묵시적 형변환

■ 묵시적 자료형의 변환

- 선언된 자료형은 대입 되는 값의 성격에 따라 언제든지 다른 자료형의 변환 가능
 - 별도의 데이터형 변환 과정이 필요 없음
- [예]
 - data라는 변수를 선언하고 200이라는 값으로 초기화
 - 변수 data에 "한국교통대학교"라는 문자열을 다시 지정할 경우

```
var data = 200;  
var data = "한국교통대학교";
```

- 변수 data는 숫자형으로 선언되고 200이라는 값을 가짐
- 기존 200이라는 값은 "한국교통대학교"라는 값으로 변경되고, data는 문자열 형으로 변환됨

연산자

연산자의 종류

■ 산술 연산자

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var num1 = 15;
    var num2 = 2;
    var result;
    result = num1 + num2;
    document.write(result, "<br>");
    result = num1 - num2;
    document.write(result, "<br>");
    result = num1 * num2;
    document.write(result, "<br>");
    result = num1 / num2;
    document.write(result, "<br>");
    result = num1 % num2;
    document.write(result, "<br>");
  </script>
</head>
<body>
</body>
</html>
```

2-operator-arithmetic.htm (p. 44)

연산자	설명
+	더하기 연산 수행
-	빼기 연산 수행
*	곱하기 연산 수행
/	나누기 연산 수행
%	나머지 연산 수행

17
13
30
7.5
1

연산자의 종류

■ 문자열 결합 연산자

- 하나 이상의 문자열을 하나의 문자열로 결합
 - 피연산자 중 하나만 문자형이라도, 나머지 피연산자들은 묵시적 형변환을 통해 문자형으로 변환됨

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <script type="text/javascript">
    var t1 = "학교종이";
    var t2 = " 땡땡땡 ";
    var t3 = 8282;
    var t4 = " 어서 모이자";
    var result;
    result = t1 + t2 + t3 + t4;
    document.write(result);
  </script>
</head>
<body>
</body>
</html>
```

2-operator-character.htm (p. 45)

연산자	설명
+	두 문자열의 결합

학교종이 땡땡땡 8282 어서 모이자

연산자의 종류

■ 대입 연산자

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var num1 = 10;
    var num2 = 3;
    num1 += num2;
    document.write(num1, "<br>");

    num1 -= num2;
    document.write(num1, "<br>");

    num1 *= num2;
    document.write(num1, "<br>");

    num1 %= num2;
    document.write(num1, "<br>");
  </script>
</head>
<body>
</body>
</html>
```

2-operator-allo.htm (p. 46)

연산자	설명
A = B	B의 값을 A의 값으로 대입
A += B	A와 B를 더한 결과를 A에 대입
A -= B	A에서 B를 뺀 결과를 A에 대입
A *= B	A와 B를 곱한 결과를 A에 대입
A /= B	A를 B로 나눈 결과를 A에 대입
A %= B	A를 B로 나눈 후 나머지 값을 A에 대입

13
10
30
0

• 대입 연산자의 예

- 여러 개의 문자형 데이터를 하나의 문자열로 결합하여 화면에 출력

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <script type="text/javascript">
    var str = "<table border='1'>";
    str += "<tr>";
    str += "<td>100</td><td>200</td><td>300</td>";
    str += "</tr>";
    str += "</table>";

    document.write( str );
  </script>

</head>
<body>
</body>
</html>
```

2-operator-allo2.htm (p. 47)

100	200	300
-----	-----	-----

연산자의 종류

■ 증감 연산자

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var num1 = 10;
    var num2 = 20;
    var result;

    num1--;
    document.write(num1, "<br>");

    num1++;
    document.write(num1, "<br>");

    result = num2++;
    document.write(result, "<br>");

    result = ++num2;
    document.write(result, "<br>");
  </script>
</head>
<body> </body>
</html>
```

2-operator-inc-dec.htm (p. 48)

연산자	설명
A++	A를 1만큼 증가시킨 후 그 결과를 다시 A에 대입
A--	A를 1만큼 감소시킨 후 그 결과를 다시 A에 대입
A = B++	B의 값을 A에 대입한 다음, B의 값을 1만큼 증가시킴
A = ++B	B를 1만큼 증가시킨 다음, B의 값을 A에 대입

9
10
20
22

연산자의 종류

■ 비교 연산자

연산자	설명
$A > B$	A가 B보다 크다.
$A < B$	A가 B보다 작다.
$A \geq B$	A가 B보다 크거나 같다.
$A \leq B$	A가 B보다 작거나 같다.
$A == B$	- A와 B가 같다. - A와 B의 자료형에 관계없이 표기된 숫자만 같으면 true를 반환. [ex] A가 숫자 10이고, B가 문자열 "10"일 경우 true를 반환
$A != B$	- A와 B는 다르다. - A와 B의 자료형에 관계없이 표기된 숫자만 같으면 false를 반환. [ex] A가 숫자 10이고, B가 문자열 "10"일 경우 false를 반환
$A === B$	- A와 B가 같다. - A와 B의 자료형에 서로 일치하고 표기된 숫자도 같을 경우 true를 반환. [ex] A가 숫자 10이고, B가 문자열 "10"일 경우 false를 반환
$A !== B$	- A와 B는 다르다. - A와 B의 자료형에 서로 일치하고 표기된 숫자도 같을 경우 false를 반환. [ex] A가 숫자 10이고, B가 문자열 "10"일 경우 true를 반환

연산자의 종류

2-operator-compare.htm (p. 51)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var a = 10;
    var b = 20;
    var c = 10;
    var f = "20";
    var result;
    result = a > b;
    document.write(result, "<br>");
    result = a < b;
    document.write(result, "<br>");
    result = a <= b;
    document.write(result, "<br>");
    result = b == f;
    document.write(result, "<br>");
    result = a != b;
    document.write(result, "<br>");
    result = b === f;
    document.write(result, "<br>");
  </script>
</head>
<body> </body>
</html>
```

```
false
true
true
true
true
false
```

연산자의 종류

■ 논리 연산자

연산자	설명
A B	A 또는 B 중 하나만 true일 경우 true를 반환 (OR연산자)
A && B	A와 B가 모두 true일 경우에만 true를 반환 (AND 연산자)
!A	A가 true일 경우 false를 반환, false일 경우 true를 반환 (NOT 연산자) (단항 연산자)

■ 연산자의 우선순위

우선순위	설명
1	()
2	단항 연산자 (--, ++, !)
3	산술 연산자 (+, -, *, /, %)
4	비교 연산자 (>, >=, <, <=, ==, ==, !=, !=)
5	논리 연산자 (&&,)
6	대입 연산자 (=, +=, -=, *=, /=, %=)

연산자의 종류

2-operator-priority.htm (p. 53)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var a = 10, b = 20, m = 30, n = 40;
    var result;

    result = a>b || b>=m;
    document.write(result, "<br>");

    result = a>b || b>=m || m<=n;
    document.write(result, "<br>");

    result = a<=b && b>=m && m<=n;
    document.write(result, "<br>");

    result = a<=b && b<=m && m<=n;
    document.write(result, "<br>");

    result= !(a>b);
    document.write(result, "<br>");
  </script>
</head>
<body> </body>
</html>
```

```
false
true
false
true
true
```

연산자의 종류

■ 삼항 조건 연산자

- 연산의 수행 결과에 따라 실행되는 스크립트 코드를 달리 하고 싶을 때 사용하는 연산자

조건식 ? 실행문1 : 실행문2 ;

- 조건식의 결과가 true이면 실행문1을 수행하고, 거짓인 경우 실행문2를 수행

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    var a = 10;
    var b = 3;
    var result = a > b ? "javascript" : "hello";
    document.write(result);
  </script>
</head>
<body>
</body>
</html>
```

2-operator-three.htm (p. 54)

javascript

■ 적정 체중 구하기

• 적정 체중 계산법

$$\text{적정 체중} = (\text{본인 신장} - 100) * 0.9$$

• 체중 계산의 예

신장 : 180cm
체중 : 74kg

$$\text{적정 체중} = (180-100) * 0.9 = 72\text{kg}$$

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var name = prompt("당신의 이름은?", "");
    var height = prompt("당신의 신장은?", "0");
    var weight = prompt("당신의 몸무게는?", "0");

    var normal_w = (height - 100) * 0.9;
    var result = weight >= normal_w - 5 && weight <= normal_w + 5; //오차범위 ±5

    result = result ? "적정 체중입니다." : "적정 체중이 아닙니다.";

    document.write(name + "님은 " + result);

  </script>
</head>
<body>
</body>
</html>
```


연산자 실습

localhost:8080 내용:

당신의 이름은?

홍길동

확인

취소

localhost:8080 내용:

당신의 신장은?

180

확인

취소

localhost:8080 내용:

당신의 몸무게는?

75

확인

취소

홍길동님은 적정 체중입니다.

수고하셨습니다.