

변수

- 데이터 타입 -

데이터 타입

❖ 자바스크립트 자료형의 종류

- 숫자
- 문자열
- 불리언
- 함수
- 객체
- 정의되지 않은 자료형

데이터 타입

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

- ❖ 정수와 실수를 구분하지 않음
- ❖ 연산자 우선순위 고려해 프로그래밍해야 함
- ❖ 나머지 연산자
 - 좌변을 우변으로 나눈 나머지

연산자	설명
%	나머지 연산자

데이터 타입

❖ 문자열

- 문자의 집합

❖ 문자열 생성 방법

- 큰따옴표 안에 문자를 넣는 방법
- 작은따옴표 안에 문자를 넣는 방법(추천)
- 문자열 안에 따옴표를 넣을 경우
 - 서로 다른 따옴표로 문자열을 감싸줌

데이터 타입

❖ 이스케이프 문자

- 특수 기능 수행

이스케이프 문자	설명	이스케이프 문자	설명
\t	수평 탭	\'	작은따옴표
\n	행 바꿈	\"	큰따옴표
\\	역 슬래시		

❖ 문자열은 문자열 연결 연산자를 사용하여 합칠 수 있음

연산자	설명
+	문자열 연결 연산자

데이터 타입

❖ 02_1_01_newline.html

```
<body>
  <script>
    document.write("토끼\n거북이");
    function btnClick() {
      alert("토끼\n거북이");
    }
  </script>
  <p>
    <input type="button" value="대화상자 열기"
      onClick="btnClick()">
  </p>
</body>
```

데이터 타입

❖ 02_1_02_backslash.html

```
<body>
  <script>
    document.write("아주 엄청나게 긴 문자열을 한 행에 입력할 때는 \
      백슬레시를 행 끝에 붙이고 다음 줄로 내려 쓸 수 있습니다.");
  </script>
</body>
```

데이터 타입

❖ 불리언

- 결과를 참과 거짓으로 표현할 때 사용

연산자	설명
>=	좌변이 우변보다 크거나 같음
<=	우변이 좌변보다 크거나 같음
>	좌변이 우변보다 큼
<	우변이 좌변보다 큼
==	좌변과 우변이 같음
!=	좌변과 우변이 다름

- 문자열도 비교 가능
 - 국어 사전의 앞쪽에 위치할수록 값이 작음

데이터 타입

❖ 논리 연산자

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

❖ 논리 부정 연산자

- 참 → 거짓, 거짓 → 참으로 바꿈

❖ 논리합 연산자

- 양쪽이 모두 거짓 → 거짓, 나머지는 참

❖ 논리곱 연산자

- 양쪽이 모두 참 → 참, 나머지는 거짓

❖ 02_1_03_boolean.html

```
<body>
  <script>
    var a = 3;
    if (a == 3) {
      document.write("a는 3입니다.");
    } else {
      document.write("a는 3이 아닙니다.");
    }
  </script>
</body>
```

변수 - 변수 -

변수

❖ 변수

- 값을 저장할 때 사용하는 식별자
- 숫자뿐만 아니라 모든 자료형 저장 가능

❖ 변수의 사용 과정

- 변수 선언 (var 키워드 사용)
 - `var a;`
- 변수 초기화 (처음 값을 할당)
 - `var a = 12;`
- 연속 선언
 - `var a, b=3, c;`

❖ 02_2_01_var.html

```
<body>
  <script>
    var a = 12;
    var b = "korea";
    document.write("a = " + a);
    document.write("<br>");
    document.write("b = " + b);
  </script>
</body>
```

변수

❖ 선언없이 사용하기

```
a = 12;  
b = 'Korean';
```

- 선언없이 사용한 변수는 전역 변수가 됨

❖ 동적 타입

- 변수의 타입은 언제든지 변경 가능

❖ 02_2_02_dynamictype.html

```
<body>
  <script>
    a = 12;
    document.write("a = " + a + "<br>");
    a = "korea";
    document.write("a = " + a);
  </script>
</body>
```

변수

❖ 식별자

- 자바스크립트에서 이름을 붙일 때 사용하는 단어

❖ 식별자의 규칙

- 키워드를 사용하면 안 된다.
- 특수문자는 _과 \$만 허용한다.
- 숫자로 시작하면 안 된다.
- 공백은 입력하면 안 된다.

❖ 식별자를 만들 때 지키는 관례

- 생성자 함수의 이름은 항상 대문자로 시작한다.
- 변수, 인스턴스, 함수, 메소드의 이름은 항상 소문자로 시작한다.
- 여러 단어로 이뤄진 식별자는 각 단어의 첫 글자를 대문자로 한다.

변수

❖ 식별자의 종류

- 변수
- 속성
- 함수
- 메소드

구분	단독으로 사용	다른 식별자와 함께 사용
식별자 뒤에 괄호 없음	변수	속성
식별자 뒤에 괄호 있음	함수	메소드

someName : 변수 xxx.someName : 속성
someName() : 함수 xxx.someName() : 메서드

변수

❖ 주석

- 프로그램의 진행에 전혀 영향을 주지 않는 코드
- 프로그램을 설명하는데 사용

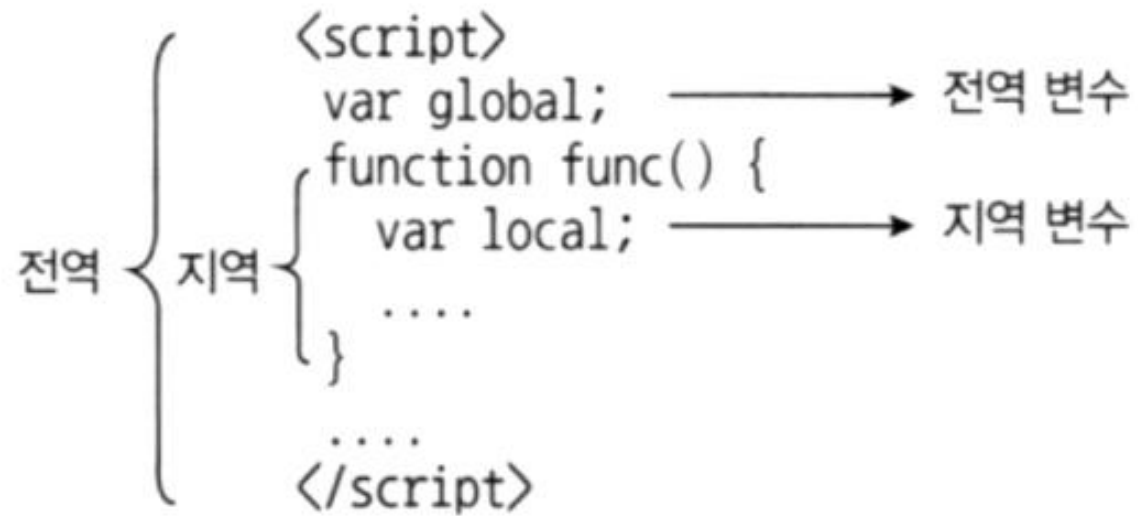
방법	표현
① 한 행 주석 처리	// 주석문
② 여러 행 주석 처리	/* 주석문 주석문 */

❖ 02_2_03_wrongname.html

```
<body>
  <script>
    radius = 3;
    pi = 3.14;
    circle = radius * pi * 2;
    document.write("원주 둘레는 " + circle + "입니다.");
  </script>
</body>
```

변수

❖ 변수의 범위



❖ 02_2_04_varscope.html

```
<body>
  <script>
    var global = "전역";
    function func() {
      var local = "로컬";
      document.write("함수안 local = " + local + "<br>");
      document.write("함수안 global = " + global + "<br>");
    }
    func();
    // document.write("함수밖 local = " + local + "<br>");
    document.write("함수밖 global = " + global + "<br>");
  </script>
</body>
```

❖ 02_2_05_global.html

```
<script>
    var global = "전역";
</script>

<body>
    <script>
        function func() {
            var local = "로컬";
            document.write("함수안 local = " + local + "<br>");
            document.write("함수안 global = " + global + "<br>");
        }
        func();
        // document.write("함수밖 local = " + local + "<br>");
        document.write("함수밖 global = " + global + "<br>");
    </script>
</body>
```

❖ 02_2_06_nameconflict.html

```
<body>
  <script>
    var score = 100;
    function func() {
      var score = 77;
      document.write("함수안 score = " + score + "<br>");
    }
    func();
    document.write("함수밖 score = " + score + "<br>");
  </script>
</body>
```

❖ 02_2_07_novar.html

```
<body>
  <script>
    var score = 100;
    function func() {
      score = 77;
      document.write("함수안 score = " + score + "<br>");
    }
    func();
    document.write("함수밖 score = " + score + "<br>");
  </script>
</body>
```


❖ 02_2_08_blockscope.html

```
<body>
  <script>
    for (var i = 0; i < 3; i++) {
      var k = 1234;
      document.write("i = " + i + "<br>");
    }
    document.write("i = " + i + " ,k = " + k + "<br>");
  </script>
</body>
```

- let 키워드
 - 블록 범위 변수 선언

변수

❖ 상수

- `const` 키워드

❖ 02_2_09_const.html

```
<body>
  <script>
    const MILE = 1.609;
    var marathon = 42.195;
    document.write("마라톤 코스 길이는 " +
                  (marathon / MILE) + "마일이다.");
  </script>
</body>
```

❖ 02_2_10_defineconst.html

```
<body>
  <script>
    const RATIO = 3.25;
    var a = 1000;
    var b = 2850;
    document.write(a + "만원 정기 예금의 1년 이자 = " +
      (a * RATIO / 100) + "만원<br>");
    document.write(b + "만원 정기 예금의 1년 이자 = " +
      (b * RATIO / 100) + "만원<br>");
  </script>
</body>
```

변수

❖ 특수한 값

- `null`
 - 아무것도 참조하지 않음을 나타냄
- `undefined`
 - 초기화 되지않은 변수가 가지는 값
- `NaN`
 - 연산결과가 숫자가 아님
- `infinite`
 - 무한대 숫자

❖ 02_2_11_undefined.html

```
<body>
  <script>
    var notinit;
    document.write("초기화 되지 않은 변수 : " + notinit);
    document.write("존재 하지 않는 변수 : " + ghost);
  </script>
</body>
```

❖ 02_2_13_nan.html

```
<body>
  <script>
    var veryBig = 1234/0;
    document.write("veryBig = " + veryBig + "<br>");

    var noNumber = Math.sqrt(-2);
    document.write("noNumber = " + noNumber);
  </script>
</body>
```

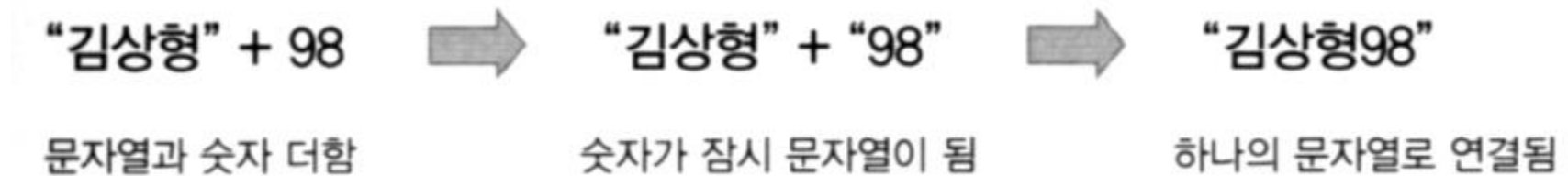
변수

- 타입 변환 -

타입 변환

❖ 암시적 변환

- + 연산자
 - 문자열이 우선
 - 문자열이 아닌 타입 → 문자열



- 다른 연산자
 - 숫자가 우선
 - 숫자 이외의 타입 → 숫자

❖ 02_3_01_implicit.html

```
<body>
  <script>
    var name = "김상형 : ";
    var score = 98;
    document.write(name + score + "<br>");

    var value1 = "8";
    var value2 = "6";
    var add = value1 + value2;
    document.write("add : " + add + "<br>");

    var sub = value1 - value2;
    document.write("subtract : " + sub + "<br>");
  </script>
</body>
```

타입 변환

❖ 명시적 변환

- Number()
 - 숫자 이외의 값이 포함되어 있으면 NaN 배정
- parseInt(), parseFloat()
 - 숫자 이외의 값이 포함되어 있으면 숫자 부분만 해석

❖ 02_3_02_explicit.html

```
<body>
  <script>
    var korean = "82";
    var english = "75";
    var total = korean + english;
    document.write("총점은 " + total + "이다.<br>");
  </script>
</body>
```

❖ 02_3_03_tonumber .html

```
<body>
  <script>
    var korean = "82";
    var english = "75";
    var total = Number(korean) + Number(english);
    document.write("총점은 " + total + "이다.<br>");
  </script>
</body>
```

타입 변환

❖ 숫자를 문자열로 변환

- `String()`

❖ 02_3_04_tostring.html

```
<body>
  <script>
    var staff = "김상형 : ";
    var salary = 320;
    document.write(staff + String(salary) + "만원");
  </script>
</body>
```

타입 변환

❖ 02_3_04_tostring2.html

```
<body>
  <script>
    var staff = "김상형 : ";
    var salary = 320;
    var bonus = 160;
    document.write(staff + salary + bonus + "만원");
  </script>
</body>
```


❖ 02_3_04_tostring3.html

```
<body>
  <script>
    var staff = "김상형 : ";
    var salary = 320;
    var bonus = 160;
    document.write(staff + String(salary + bonus) + "만원");
  </script>
</body>
```

타입 변환

❖ 논리형 변환

- false로 해석하는 경우
 - 0
 - ""
 - null
 - undefined
 - NaN
- 나머지는 true로 해석
- 변환함수
 - Boolean()

타입 변환

❖ 02_3_05_toboolean.html

```
<body>
<script>
  if (0) {
    document.write("참입니다.");
  } else {
    document.write("거짓입니다.");
  }
</script>
</body>
```