훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 변수명 규칙

- **변수명 작성 규칙**
  - 문자, 밑줄(_), 달러 기호($)로 시작
  - 대소문자 구별('변수 A'와 '변수 a'는 서로 다른 변수)
  - 한글은 사용 가능하나 영문자 사용 권장
  - 자바스크립트에서 정한 예약어는 변수명으로 사용 불가능

abstract	Arguments	boolean	break	byte
case	catch	char	class	const
continue	debugger	default	delete	do
double	else	enum	eval	export
extends	false	final	finally	float
for	function	goto	if	implements
import	in	instanceof	int	interface
let	long	native	new	null
package	private	protected	public	return
short	static	super	switch	synchronized
this	throw	throws	transient	true
try	typeof	var	void	volatile
while	with	yield		

✤ **Javascript 예약어**

❖ 변수 사용 방법

● 변수 사용 예

- var x; // 변수 x 선언
- var y=10; // 변수 y 선언 및 초깃값 할당
- var x=y; // 변수 y의 값을 변수 x에 저장
- var a, b, c; // 변수 a, b, c 선언
- var a=10, b=11, c=12; // 변수 a, b, c 선언 및 각각 다른 초깃값 할당
- var a=b=c=10; // 변수 a, b, c 선언 및 같은 초깃값 할당
- var name="홍길동", age=25; // 변수 name, age 선언 및 각각 다른 초깃값 할당
- var total=a+b+c; // 변수 a, b, c 값을 더한 결과를 변수 total에 저장

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 변수 사용 방법

● 변수 사용 시 문법적으로 오류가 발생한 사례

- var 7num=100; // 숫자로 시작하는 변수명 잘못 사용
- var &num=100; // 특수 문자로 시작하는 변수명 잘못 사용
- var true=1; // 예약어를 변수명으로 잘못 사용
- var 10=x; // 좌변에 상수값 잘못 선언
- var a+b=20; // 좌변에 연산식 잘못 선언
- var “홍길동”=name; // 좌변에 문자열값 잘못 선언
- var get Number=100; // 변수명 사이에 공백(space) 잘못 선언
- var a, b, c=100; // 콤마로 구분한 변수명 잘못 선언

❖ Javascript 포함 방법

◆ 변수의 재선언후 데이터 타입

js_data_retype02.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
</head>
<body>
    <script>
        stdName="홍길동";    // var 키워드 생략
        comGrade=96;         // var 키워드 생략
        var stdName;         // 변수명 재선언
        var comGrade;        // 변수명 재선언
        document.write("학생 이름 : " + stdName + "<br>");
        document.write("컴퓨터 점수 : " + comGrade + "<br>");
    </script>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 전역변수 & 지역변수

- **전역 변수**
  - 코드 내 어느 위치에서든 선언하여 전 영역에서 사용할 수 있는 변수
- **지역 변수**
  - 변수가 선언된 해당 블록에서 선언하여 범위 내에서만 유효하게 사용할 수 있는 변수

js_var01.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <title>varible</title>
    <script type="text/javascript">
        var globValue1;
        globValue2;
        function test01() {
                var locValue;          // 지역변수
                globValue = 999;                // 함수 내부에서 var생략할경우 자동 전역변수
                locValue = 10;         // 지역변수
        }
        test01();   // 함수 실행
        alert(globValue);
    </script>
</head>
<body>
</body>
</html>
```

전역 변수와 지역 변수

❖ 변수 호이스팅 (hoisting)

- **JavaScript 변수의 특이한 점은 예외를 받지 않고도, 나중에 선언된 변수를 참조할 수 있다는 것**
  - JavaScript 변수가 어떤 의미에서 "끌어올려지거"나 함수나 문의 최상단으로 올려지는 것 (Hoisting)
  - 끌어올려진 변수는 undefined 값을 반환
    * 이 변수를 사용 혹은 참조한 후에 선언 및 초기화하더라도, 여전히 undefined를 반환

`js_var02.html`

```
console.log(x === undefined); // logs "true"
var x = 3;

// undefined 값을 반환함.
var myvar = "my value";
(function() {
  console.log(myvar); // undefined
  var myvar = "local value";
})();
```

```
var x;
console.log(x === undefined); // logs "true"
x = 3;

var myvar = "my value";
(function() {
  var myvar;
  console.log(myvar); // undefined
  myvar = "local value";
})();
```

✻ 호이스팅 때문에, 함수 내의 모든 var 문은 가능한 함수 상단 근처에 두는 것이 좋다.
이 방법은 코드를 더욱 명확하게 만들어줍니다.

❖ 전역 변수와 지역 변수 이해

js_var03.html

```
<script>
   function getGrade() {    // 함수 정의
            var kor=95;          // 지역 변수
   }
   var kor=100;            // 전역 변수
   getGrade();            // 함수 호출
   document.write("국어 점수 : " + kor + "<br>");
</script>
```

js_var04.html

```
<script>
   function getGrade() {    // 함수 정의
      kor=95;              // 자동 전역 변수
   }
   var kor=100;             // 전역 변수
   getGrade();              // 함수 호출
   document.write("국어 점수 : " + kor + "<br>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 전역 변수와 지역 변수 이해

js_var05.html

```
<script>
    function getGrade() {  // 함수 정의
        var kor=95;        // 지역 변수
    }
    getGrade();            // 함수 호출
    document.write("지역 변수 값은 함수 외부에서 사용할 수 없습니다.<br>");
    document.write("국어 점수 : " + kor + "<br>");
</script>
```

js_var06.html

```
<script>
    function getGrade() {   // 함수 정의
            var kor=95;         // 지역 변수
            return kor;
    }
    getKor=getGrade();      // 함수 호출
    document.write("국어 점수 : " + getKor + "<br>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 전역 변수

- **전역 변수는 global 객체의 속성(property)**

  - 웹 페이지에서 global 객체는 window 이므로,
    windows.variable 구문을 통해 전역 변수를 설정하고 접근할 수 있다.

  - window 혹은 frame의 이름을 지정하여
    한 window 혹은 frame에서 다른 window 혹은 frame에 선언된 전역 변수에 접근할 수 있다.

    예] memberNo 라는 변수가 문서에 선언된 경우,
    iframe에서 parent.memberNo로 이 변수를 참조할 수 있다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 전역 변수

member.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <title>varible</title>
    <link rel="stylesheet" type="text/css" href="./CSS/style.css" />
    <script type="text/javascript">
        var memberNo = 777;
    </script>
</head>
<body>
    <div class="card">
        <h3>여기는 메인 페이지</h3>
    </div>
    <iframe class="card" src="frame01.html" />
</body>
```

frame01.html

```
<!DOCTYPE html>
<html lang="UTF-8">
<head>
    <meta charset="UTF-8" />
    <title>varible</title>
    <script type="text/javascript">
        var mNo = parent.memberNo;
        document.write("여기는 iframe_page : 회원번호는 \"" + mNo + "\"");
    </script>
    <style type="text/css">
        body { border: 0px; text-align: center;}
    </style>
</head>
<body>
</body>
```

./CSS/style.css

```
html, body {
    margin: 0px;
    text-align: center;
}
.card {
    box-shadow: 0px 0px 7px gray;
    border:1px solid gray;
    width: 330px;
}
div {
    background-color: ivory;
    text-align: center;
    height: 50px;
    margin : 20px auto 10px;
}
h3 {
    line-height: 15px;
}
iframe {
    background-color: yellow;
    margin-top: 0px;
    margin-bottom: 0px;
}
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 연산자

- ✤ 피연산자에게 연산 명령을 내리기 위해 사용하는 기호

❖ 연산자의 종류

연산자	기호
문자열 연산자	+(문자열 연결)
산술 연산자	++(증가 연산), --(감소 연산), *(곱셈), /(나눗셈), %(나머지), +(덧셈), -(뺄셈)
비교 연산자	<(작다), <=(작거나 같다), >(크다), >=(크거나 같다), ==(값이 같다), !=(값이 다르다), ===(값과 타입 모두 같다), !==(값 또는 타입이 다르다)
논리 연산자	&(비트 AND), |(비트 OR), ^(비트 XOR), &&(논리 AND), ||(논리 OR)
조건 연산자	(판단) ? true : false;
대입 연산자	=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, |=, ^=

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 문자열 연산자

- ‘+’ 기호를 사용하여 문자열을 연결

```
var st = “Hello” + “Javascript”;          // 연산결과 “Hello Javascript”가 출력
```

```
var st = “100” + 10; // 10010 출력
var st = 100 + 10 ;          // 110   출력
```

❖ 산술 연산자

- 사칙 연산을 수행
- 종류 : 더하기(+), 빼기(-), 곱하기(*), 나누기(/), 나머지(%), 증감(++), 감소(--)
  - 나머지(%) : 나눗셈 결과 나머지 값을 구함
  - 증가(++) : 변숫값을 증가시킴
  - 감소(--) : 변숫값을 감소시킴

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 비교 연산자

- 두 피연산자의 값을 비교하여 참(true) 또는 거짓(false) 값을 반환

비교 연산자	설명	사용 예	결과
==	값이 같은지 비교한다.	x=="5"	true
===	값과 타입이 같은지 비교한다.	x==="5"	false
!=	값이 다른지 비교한다.	x!="5"	false
!==	값 또는 타입이 다른지 비교한다.	x!=="5"	true

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 논리 연산자

* 일반 논리 연산자

논리곱(&&)	두 개의 피연산자 값이 모두 참일 때만 참이고, 하나라도 거짓이면 거짓
논리합*(||)*	두 개의 피연산자 값 중 하나라도 참이면 참이고, 모두 거짓이면 거짓
논리 부정*(!)*	피연산자 값이 참이면 거짓, 거짓이면 참

❖ 대입 연산자

* ‘=’ 기호를 사용하여 값이나 변수를 할당

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 조건(삼항) 연산자

* 조건식을 판별하여 참이냐 거짓이냐에 따라 다음 문장을 선택적으로 실행

```
조건 ? 값1 : 값2
```

❖ 단항 연산자

* **delete : 객체, 객체의 속성 또는 배열의 특정한 위치에 있는 객체를 삭제**
* **typeof : 데이터의 자료형 반환**
* **instanceof : 명시된 객체가 명시된 객체형인 경우 true를 반환**

❖ this

* **현재 객체를 참조하는 데 this 키워드를 사용**
  : 일반적으로, this는 함수에서 호출하는 객체를 참조