



2. UI 구현하기(2001020225_16v4.2)

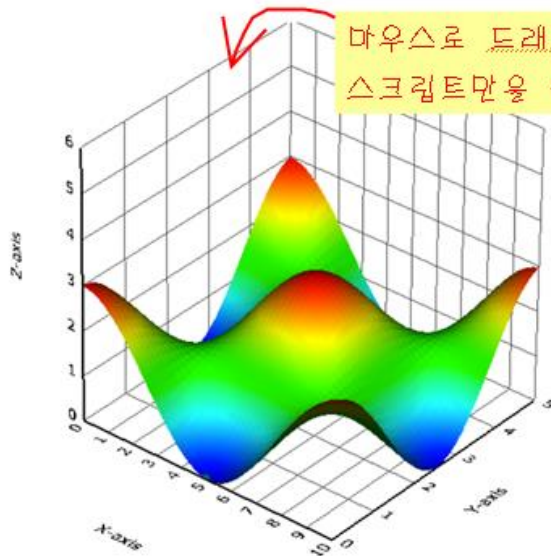
자바 스크립트 기초



2. UI 구현하기(2001020225_16v4.2)

자바 스크립트 소개

- 자바스크립트(javascript): 동적인 웹 페이지를 작성하기 위하여 사용되는 언어
- 웹의 표준 프로그래밍 언어
- 모든 웹브라우저들은 자바스크립트를 지원



마우스로 드래그하여 회전할 수 있는 이러한 화면도 자바스크립트만을 이용하여 구현되었다.



2. UI 구현하기(2001020225_16v4.2)

HTML5 기술의 핵심





2. UI 구현하기(2001020225_16v4.2)

자바 vs 자바 스크립트

특징	자바 언어	자바스크립트
언어 종류	소스 파일을 컴파일하여 실행하는 컴파일 언어이다.	브라우저가 소스 코드를 직접 해석하여 실행하는 인터프리트 언어이다.
실행 방식	자바 가상 기계 위에서 실행한다.	브라우저 위에서 실행된다.
작성 위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수 선언	변수의 타입을 반드시 선언해야 함	변수의 타입을 선언하지 않아도 사용 가능



2. UI 구현하기(2001020225_16v4.2)

자바 스크립트 역사

- 넷스케이프의 브렌던 아이크(Brendan Eich)가 개발
- 처음에는 라이브스크립트(LiveScript)
- 최신 버전은 자바스크립트 1.8.5
- ECMA(European Computer Manufacturer's Association)이 ECMAScript라는 이름으로 표준을 제정-> ECMA-262





2. UI 구현하기(2001020225_16v4.2)

자바 스크립트 특징

- 인터프리트 언어
- 동적 타이핑(dynamic typing)
- 구조적 프로그래밍 지원
- 객체 기반
- 함수형 프로그래밍 지원
- 프로토타입-기반(prototype-based)



2. UI 구현하기(2001020225_16v4.2)

자바 스크립트 특징

키워드

키워드란?

자바스크립트가 처음 만들어질 때 정해진 특별한 의미가 있는 단어
모든 브라우저에서 표와 같은 **28**개의 키워드를 지원

자바스크립트의 키워드(1)

break	else	instanceof	true
case	false	new	try
catch	finally	null	typeof
continue	for	return	var
default	function	switch	void
delete	if	this	while
do	in	throw	with



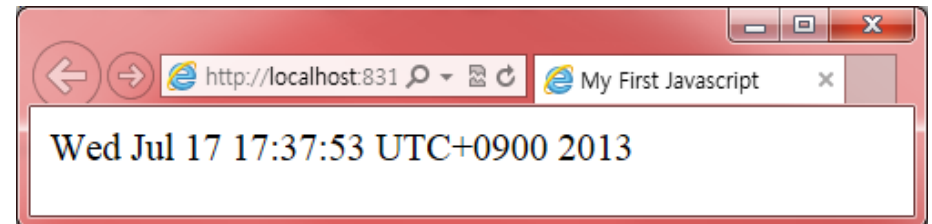
2. UI 구현하기(2001020225_16v4.2)

첫 번째 예제

```
<!DOCTYPE HTML>
<html>
<head>
  <title> My First Javascript </title>
</head>
<body>

  <script>
    var now = new Date();
    document.write(now);
  </script>

</body>
</html>
```



웹브라우저에서 실행



2. UI 구현하기(2001020225_16v4.2)

자바 스크립트의 용도

- 이벤트에 반응하는 동작을 구현
- AJAX
- HTML 요소들의 크기나 색상을 동적으로 변경
- 게임이나 애니메이션
- 사용자가 입력한 값들을 검증



2. UI 구현하기(2001020225_16v4.2)

자바 스크립트의 위치

- 내부 자바스크립트
- 외부 자바스크립트
- 인라인 자바스크립트



2. UI 구현하기(2001020225_16v4.2)

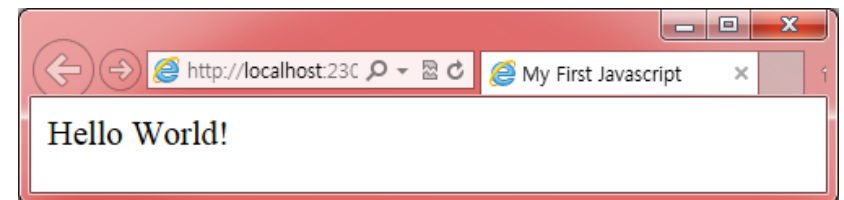
내부 자바 스크립트

```
<!DOCTYPE HTML>
<html>
<head>

  <title>My First Javascript </title>

  <script>
    document.write("Hello World!");
  </script>

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





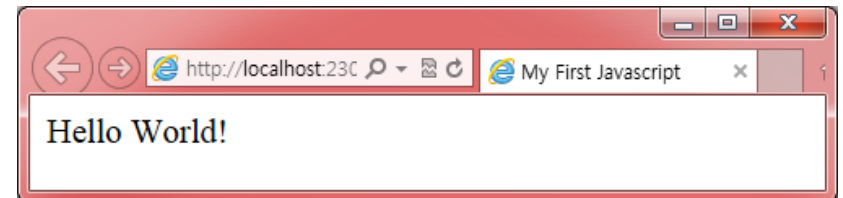
2. UI 구현하기(2001020225_16v4.2)

외부 자바 스크립트

```
<!DOCTYPE html>
<html>
<head>
  <script src="myscript.js"></script>
</head>
<body>
</body>
</html>
```

myscript.js

```
document.write("Hello World!");
```





2. UI 구현하기(2001020225_16v4.2)

인라인 자바 스크립트

```
<!DOCTYPE html>  
<html>  
<body>  
  <button type="button" onclick="alert('반갑습니다.')">버튼을 누르세요!</button>  
</body>  
</html>
```





2. UI 구현하기(2001020225_16v4.2)

문장

- 자바스크립트 문장(statement)들은 웹 브라우저에게 내리는 명령

1. 문서에 "Hello World!"를 추가하시오.
2. 화면에 경고창을 띄우시오.
3. 변수를 하나 만드시오.
4. ...



```
document.write("Hello World!");  
alert("warning!!");  
var count;  
...
```



순차적
으로
실행된
다.



2. UI 구현하기(2001020225_16v4.2)

변수

◆식별자

자바스크립트에서 이름을 붙일 때 사용

식별자의 예

변수명과 함수명

식별자 생성 시 규칙

키워드를 사용 불가

숫자로 시작하면 불가

특수 문자는 _과 \$만 허용

공백 문자 포함 불가

E>

```
alpha
alpha10
_alpha
$alpha
AlPha
ALPHA
```

식별자로 사용 불가능한 단어

```
break
273alpha
has space
```



2. UI 구현하기(2001020225_16v4.2)

변수

❖ 식별자 생성 규칙

모든 언어가 사용 가능하나 알파벳 사용이 개발자들 사이 관례

Input, output 같은 의미 있는 단어 사용

자바 스크립트 개발자가 식별자를 만들 때 지키는 관례

생성자 함수의 이름은 대문자로 시작

변수와 인스턴스, 함수, 메서드의 이름은 항상 소문자로 시작

식별자가 여러 단어로 이루어지면 각 단어의 첫 글자는 대문자

Ex)

will out	⇒ willOut
will return	⇒ willReturn
i am a boy	⇒ iAmABoy



2. UI 구현하기(2001020225_16v4.2)

변수

- 변수(**variable**)는 데이터를 저장하는 상자
- var 키워드를 사용하여 선언(declare)한다.

변수 이름이다. 스크립트 안에서 유일해야 한다.

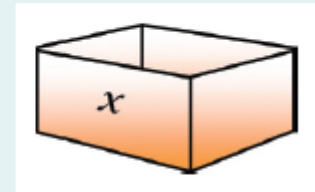
var

x

;

var 키워드는 변수가 선언된다는 것을 나타낸다.

세미콜론은 하나의 문장이 끝났음을 나타낸다.





2. UI 구현하기(2001020225_16v4.2)

변수

❖ 변수 할당

- 변수 초기화 : 변수 선언 후 처음 값을 할당 하는 것

코드 : 변수의 선언과 할당

```
<script>
    // 변수를 선언합니다.
    var pi;

    // 변수에 값을 할당합니다.
    pi = 3.14159265;
</script>
```

코드 : 변수의 선언과 초기화

```
<script>
    // 변수를 선언 및 초기화합니다.
    var pi = 3.14159265;
</script>
```



2. UI 구현하기(2001020225_16v4.2)

변수

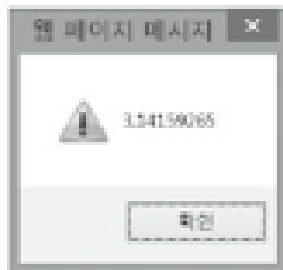
❖ 변수 할당

- 변수 사용(1)

코드 | 변수의 사용(1)

```
<script>  
    // 변수를 선언하고 초기화합니다.  
    var pi = 3.14159265;  
  
    // 출력합니다.  
    alert(pi);  
</script>
```

그림 | 변수 pi





2. UI 구현하기(2001020225_16v4.2)

변수

❖ 변수 할당

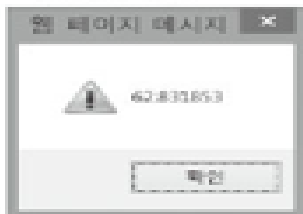
- 변수 사용(2)
 - ✓ 숫자가 들어간 변수는 숫자와 관련된 연산자 사용
 - ✓ 문자열이 들어간 변수는 문자열과 관련된 연산자 사용

코드 변수의 사용(2)

```
<script>
  // 변수를 선언 및 초기화합니다.
  var radius = 10;
  var pi = 3.14159265;

  // 출력합니다.
  alert(2 * pi * radius);
</script>
```

그림: 변수 계산





2. UI 구현하기(2001020225_16v4.2)

변수

❖ 변수에 값 할당하기

- 자료형
 - ✓ 문자열, 숫자, 불리언과 같은 것
 - ✓ 자바스크립트에는 총 여섯 가지 자료형이 있음

Cf. undefined 자료형

선언되지 않거나 할당되지 않은 변수
변수에 저장해도 의미가 없음

코드 자바스크립트의 자료형

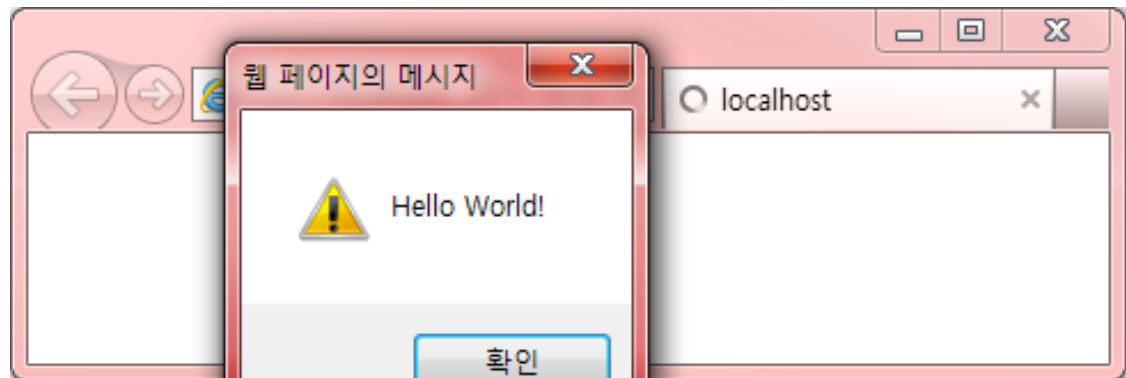
```
<script>
  // 변수를 선언합니다.
  var stringVar = 'String';
  var numberVar = 273;
  var booleanVar = true;
  var functionVar = function () { };
  var objectVar = {};
</script>
```



2. UI 구현하기(2001020225_16v4.2)

예제

```
<script>  
  var x;  
  x = "Hello World!";  
  alert(x);  
</script>
```





2. UI 구현하기(2001020225_16v4.2)

전역변수

지역변수와 전역변수

지역변수	전역변수
함수 내에 var 키워드로 선언	함수 밖에 선언되거나, 함수 내에 var 키워드 없이 선언된 변수
선언된 함수 내에서만 사용	프로그램 전역에서 사용

```
var x;    // 전역변수 x
function f() {
  var y;  // 지역변수 y 선언
  x = 10; // 전역 변수 x에 10 저장
  y = 10; // 지역 변수 y에 10 저장
  z = 10; // 새로운 전역변수 z 선언. 10으로 초기화
}
```



2. UI 구현하기(2001020225_16v4.2)

전역변수

this로 전역변수 접근

- 지역 변수와 전역 변수의 이름을 같을 때
 - 전역 변수에 접근하고자 할 때 : **this.전역변수**

```
var x;    // 전역변수
function f() {
    var x;    // 지역변수
    x = 1;    // 지역변수 x에 1 저장
    this.x = 100; // 전역변수 x에 100 저장
}
```




2. UI 구현하기(2001020225_16v4.2)

전역변수

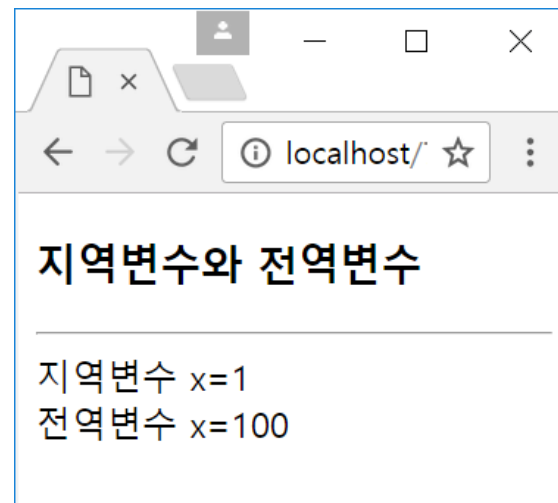
예제 지역변수와 전역변수

```
<!DOCTYPE html>
<html>
<head>
<title>지역변수와 전역변수</title></head>
<body>
<h3>지역변수와 전역변수</h3>
<hr>
<script>
var x=100; // 전역변수 x

function f() { // 함수 f() 선언
    var x=1; // 지역변수 x

    document.write("지역변수 x=" + x);
    document.write("<br>");
    document.write("전역변수 x=" + this.x);
}

f(); // 함수 f() 호출
</script>
</body>
</html>
```





2. UI 구현하기(2001020225_16v4.2)

주석

● 주석

프로그램 진행에 영향을 끼치지 않음
코드의 특정 부분을 설명

1) HTML 태그 주석

<!-- -->로 문자열을 감싸 생성

HTML 주석

```
<!DOCTYPE html>
<html>
<head>
  <!-- 주석입니다. -->
  <script>
  </script>
</head>
<body>
  <!-- <h1>주석입니다.</h1> -->
</body>
</html>
```



2. UI 구현하기(2001020225_16v4.2)

주석

◆ 주석

2) 자바스크립트 주석

//를 사용해 한 줄 주석 표현

// 뒤의 문장은 실행되지 않음

Ex) // 주석문

/*와 */을 사용해 여러 줄 주석 표현

/*와 */ 사이의 문장은 실행되지 않음

Ex) /*

주석문

주석문

*/

자바스크립트 주석

```
<script>
  // 주석은 코드의 실행에 아무 영향을 미치지 않습니다.
  /*
    alert('Hello JavaScript');
    alert('Hello JavaScript');
    alert('Hello JavaScript');
  */
</script>
```



2. UI 구현하기(2001020225_16v4.2)

자료형

- 수치형(number)
- 문자열(string)
- 부울형(boolean)
- 객체형(object)
- undefined



2. UI 구현하기(2001020225_16v4.2)

출력

❖ 자바스크립트 출력

기본 출력 방법 : **alert ()** 함수를 사용할 구성 예제 코드

코드 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script>

  </script>
</head>
<body>

</body>
</html>
```



2. UI 구현하기(2001020225_16v4.2)

출력

❖ 자바스크립트 출력

Alert () 함수

가장 기본적인 출력 방법

브라우저에 경고창을 띄울 수 있음

alert () 함수의 사용 예 :

함수의 괄호 안에는 문자열 입력

코드 기본적인 출력

```
<script>  
    alert('Hello JavaScript..!');  
</script>
```



2. UI 구현하기(2001020225_16v4.2)

출력 함수

자바스크립트로 HTML 콘텐츠 출력

- 자바스크립트로 HTML 콘텐츠를 웹 페이지에 직접 삽입
 - 바로 브라우저 윈도우에 출력
 - `document.write()`
예) `document.write("<h3>Welcome!</h3>");`
 - `document.writeln()`
 - `writeln()`은 텍스트에 '`\n`' 을 덧붙여 출력
 - '`\n`'을 덧붙이는 것은 고작해야 빈칸 하나 출력
 - 다음 줄로 넘어가는 것은 아님

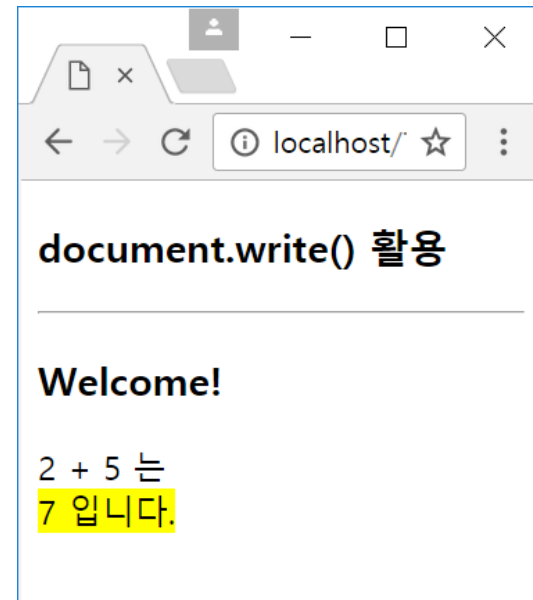


2. UI 구현하기(2001020225_16v4.2)

출력 함수

예제 document.write()로 웹 페이지에 HTML 콘텐츠 출력

```
<!DOCTYPE html>
<html>
<head><title>document.write() 활용</title>
</head>
<body>
<h3>document.write() 활용</h3>
<hr>
<script>
  document.write("<h3>Welcome!</h3>");
  document.write("2 + 5 는 <br>");
  document.write("<mark>7 입니다.</mark>");
</script>
</body>
</html>
```





2. UI 구현하기(2001020225_16v4.2)

문자열

❖ 문자열이란?

- 문자를 표현할 때 사용하는 자료의 형태
- `alert()` 함수의 매개 변수로 쓰인 `'Hello JavaScript..!'` 와 같은 자료
- 문자열을 만드는 방법
 - 큰따옴표의 사용
 - "동해물과 백두산이"
 - 작은따옴표의 사용
 - '동해물과 백두산이'
 - 두 가지 방법 중 어떤 문자열로도 사용 가능하지만 일관되게 사용해야 함



2. UI 구현하기(2001020225_16v4.2)

수치형

숫자 자료형

- 정수와 유리수의 구분 없이 숫자는 모두 숫자

Ex)

273
52.273

- 문자열과 마찬가지로 **alert()** 함수의 괄호 안에 사용해 출력

코드 숫자

```
<script>  
  alert(273);  
  alert(52.273);  
</script>
```



2. UI 구현하기(2001020225_16v4.2)

수치형

❖ 숫자를 사용한 기본적인 사칙 연산

- 표의 연산자를 사용해 기본적인 사칙연산 가능

표: 사칙 연산자

연산자	설명	연산자	설명
+	더하기 연산자	*	곱하기 연산자
-	빼기 연산자	/	나누기 연산자

- 자바스크립트에서는 연산자 우선순위 고려함
다음 그림의 결과 값은?

```
5 + 3 * 2
```

- 덧셈을 먼저 실행하고 싶다면 괄호를 사용

```
(5 + 3) * 2
```



2. UI 구현하기(2001020225_16v4.2)

수치형

❖ 숫자를 사용한 기본적인 사칙 연산

- 자바스크립트 코드를 이용한 연산

코드 사칙 연산자와 연산자 우선 순위

```
<script>
  alert(5 + 3 * 2);
  alert((5 + 3) * 2);
</script>
```

- 출력 결과
그림의 왼쪽이 $5 + 3 * 2$ 의 결과/ 오른쪽이 $(5 + 3) * 2$ 의 결과

그림 사칙 연산자와 연산자 우선 순위





2. UI 구현하기(2001020225_16v4.2)

수치형

❖ 숫자를 사용한 기본적인 사칙 연산

- 나머지 연산자 : %

좌변을 우변으로 나눈 나머지를 표시하는 연산자

표 : 나머지 연산자

연산자	설명
%	나머지 연산자

✓ 사용 예 :

코드 : 나머지 연산자

```
<script>  
    alert(10 % 7);  
</script>
```

출력 결과

그림 : 나머지 연산자





2. UI 구현하기(2001020225_16v4.2)

연산자

연산자	설명	예	수식의 값
+	덧셈	$x=3+2$	5
-	뺄셈	$x=3-2$	1
*	곱셈	$x=3*2$	6
/	나눗셈	$x=3/2$	1.5
%	나머지	$x=3\%2$	1
++	증가	$++x$	x의 값 3 → 4
--	감소	$--x$	x의 값 3 → 2



2. UI 구현하기(2001020225_16v4.2)

비교 연산자

❖ 비교 연산자

- 비교 연산자란?
두 대상을 비교할 수 있는 연산자
- 비교 연산자의 종류

표 비교 연산자

연산자	설명
>=	좌변이 우변보다 크거나 같습니다.
<=	우변이 좌변보다 크거나 같습니다.
>	좌변이 큼니다.
<	우변이 큼니다.
==	좌변과 우변이 같습니다.
!=	좌변과 우변이 다릅니다.

- 문자열 비교
국어사전의 앞 쪽에 있을 수록 값이 작음

'가방' > '하마' ⇨ false



2. UI 구현하기(2001020225_16v4.2)

비교 연산자

❖ 자바 스크립트 논리연산자의 종류

표 논리 연산자

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

- 논리부정 연산자
참을 거짓으로, 거짓을 참으로 바꿈

코드 논리 부정 연산자

```
<script>  
    alert(!true);  
    alert(!false);  
</script>
```




2. UI 구현하기(2001020225_16v4.2)

비교 연산자

❖ 비교 연산자와 논리 연산자의 적절한 사용 필요

코드 비교 연산자와 논리 연산자

```
<script>  
    alert(30 > 20 && 20 > 10);  
</script>
```

그림 표현식 '30 > 20 && 20 > 10'의 연산 과정

```
alert(30>20 && 20>10);  
    ↓  
alert(true && true);  
    ↓  
alert(true);
```

그림 범위 연산





2. UI 구현하기(2001020225_16v4.2)

복합 대입 연산자

❖ 복합 대입 연산자

- 대입 연산자와 다른 연산자를 함께 사용하는 연산자

표 1 복합 대입 연산자

연산자	설명
<code>+=</code>	기존 변수의 값에 값을 더합니다.
<code>-=</code>	기존 변수의 값에 값을 뺍니다.
<code>*=</code>	기존 변수의 값에 값을 곱합니다.
<code>/=</code>	기존 변수의 값에 값을 나눕니다.
<code>%=</code>	기존 변수의 값에 나머지를 구합니다.



2. UI 구현하기(2001020225_16v4.2)

복합 대입 연산자

❖ 복합 대입 연산자의 사용

- 변수 **value**를 10으로 초기화
- 이후 **+=** 복합 대입 연산자를 사용해 **value**의 기존 값에 10을 더함
- 결과는 20 출력

코드복합 대입 연산자

```
<script>
    // 변수를 선언합니다.
    var value = 10;

    // 연산자를 사용합니다.
    value += 10;

    // 출력합니다.
    alert(value);
</script>
```



2. UI 구현하기(2001020225_16v4.2)

복합 대입 연산자

❖ 증감 연산자

- 복합 대입 연산자를 간략하게 사용한 형태

표 : 증감 연산자

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



2. UI 구현하기(2001020225_16v4.2)

복합 대입 연산자

❖ 증감 연산자의 활용 (1)

- 변수 **number**를 초기화하고 ++ 증감 연산자 사용
 - ✓ 코드를 실행하면 10에 1을 더한 11이 출력
 - ✓ ++ number로도 출력해봐도 차이가 없음
 - ✓ 한 줄에 독립적 증감 연산자를 사용할 때는 전위와 후위의 차이가 없음
 - ✓ 다른 연산자나 함수와 함께 사용할 때 차이가 있음

코드 :증감 연산자(1)

```
<script>
    // 변수를 선언합니다.
    var number = 10;

    // 연산자를 사용합니다.
    number++;

    // 출력합니다.
    alert(number);
</script>
```



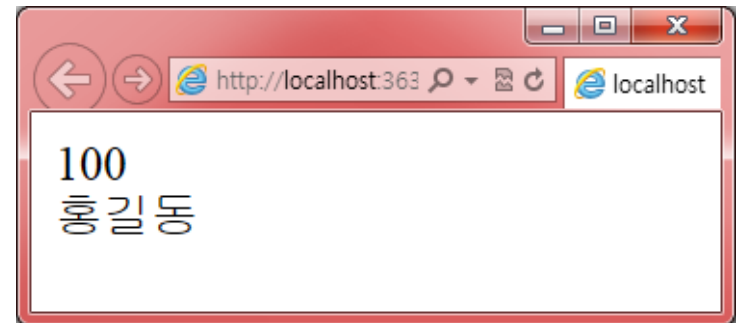
2. UI 구현하기(2001020225_16v4.2)

예제

```
<script>
  var s;

  s = 100;
  document.write(s + "<br>");

  s = "홍길동";
  document.write(s + "<br>");
</script>
```



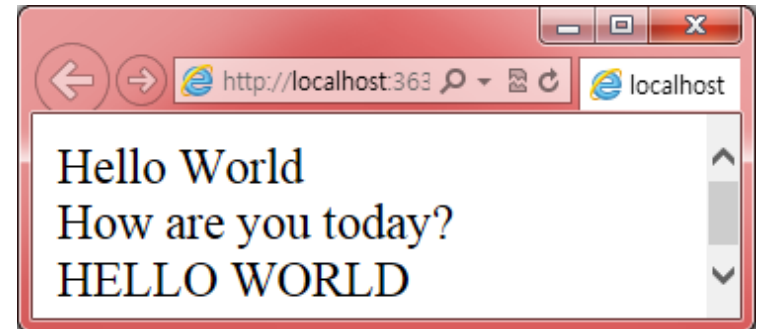


2. UI 구현하기(2001020225_16v4.2)

예제

```
<script>
  var s = "Hello World";
  var t = "How are you" + " today?";

  document.write(s + "<br>");
  document.write(t + "<br>");
  document.write(s.toUpperCase() + "<br>");
</script>
```

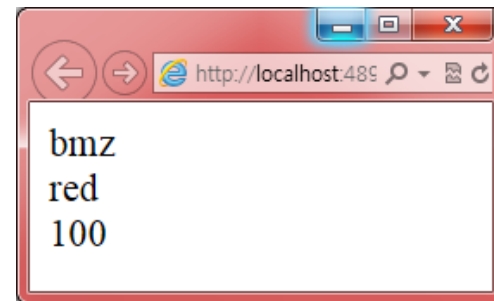
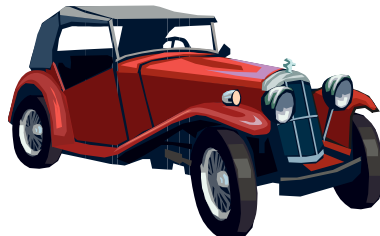




2. UI 구현하기(2001020225_16v4.2)

객체형

- 객체(**object**)는 사물의 속성과 동작을 묶어서 표현하는 기법
- (예) 자동차는 메이커, 모델, 색상, 마력과 같은 속성도 있고 출발하기, 정지하기 등의 동작도 가지고 있다.
- `var myCar = { model: "bmz", color: "red", hp: 100 };`
- `document.write(myCar.model + "
");`
- `document.write(myCar.color + "
");`
- `document.write(myCar.hp + "
");`





2. UI 구현하기(2001020225_16v4.2)

연산자

연산자	설명	예	수식의 값
+	덧셈	$x=3+2$	5
-	뺄셈	$x=3-2$	1
*	곱셈	$x=3*2$	6
/	나눗셈	$x=3/2$	1.5
%	나머지	$x=3\%2$	1
++	증가	$++x$	x의 값 3 → 4
--	감소	$--x$	x의 값 3 → 2



2. UI 구현하기(2001020225_16v4.2)

입력

❖ 문자열을 입력하는 방법

- 숫자를 입력 받는 방법
 - ✓ 문자열을 입력 받은 후 숫자로 변환
 - ✓ 문자열을 입력을 할 때 사용하는 함수는 `prompt()` – 매개변수 두 개 필요

그림 `prompt()` 함수

```
prompt()  
String prompt([String message], [String defaultValue])
```

코드 `prompt()` 함수

```
<script>  
  // 변수를 선언합니다.  
  var input = prompt('Message', 'DefStr');  
  
  // 출력합니다.  
  alert(input);  
</script>
```



2. UI 구현하기(2001020225_16v4.2)

입력

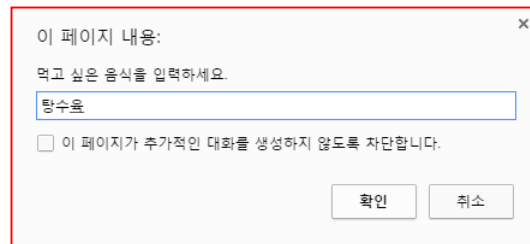
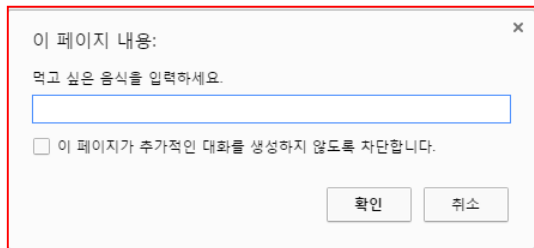
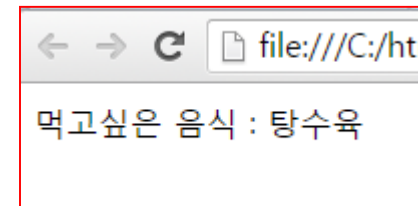
❖ Prompt() 함수 실행

- 입력 칸에 변수를 입력하면 그대로 코드 변경

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>prompt</title>
<script>
  var input = prompt("먹고 싶은 음식을 입력하세요.");
  alert(input);
  document.write("먹고싶은 음식 : " +input);

</script>
</head>

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





2. UI 구현하기(2001020225_16v4.2)

prompt() 함수

```
<script>  
  var age = prompt("나이를 입력하세요", "만나이로 입력합니다.");  
</script>
```





2. UI 구현하기(2001020225_16v4.2)

❖ 숫자와 문자열 자료형 변환

❖ prompt() 함수

- 문자열만 입력 가능
- 코드는 아무리 숫자를 입력해도 문자열의 자료형 출력

코드 숫자 입력

```
<script>
  // 변수를 선언합니다.
  var input = prompt('숫자를 입력해주세요.', '숫자');

  // 출력합니다.
  alert(typeof (input));
</script>
```



2. UI 구현하기(2001020225_16v4.2)

예 제

```
<script>
  var x, y;
  var input;

  input = prompt("정수를 입력하시오", "정수로");
  x = parseInt(input);

  input = prompt("정수를 입력하시오", "정수로");
  y = parseInt(input);

  document.write(x + y + "<br>");
</script>
```



웹브라우저에서 실행



2. UI 구현하기(2001020225_16v4.2)

예 제

```
<html>
<head>

  <title>Calclator</title>
  <script>
    function calc() {

      var x = document.getElementById("x").value;

      var y = document.getElementById("y").value;
      var sum;

      sum = parseInt(x) + parseInt(y);
      document.getElementById("sum").value = sum;
    }
  </script>
</head>
```

웹브라우저에서 실행



2. UI 구현하기(2001020225_16v4.2)

예제

```
<body>
  <h3>덧셈 계산기</h3>
  <form name="myform" action="..." method="POST">
    첫째 정수:
    <input id="x" /><br />
    두번째 정수:
    <input id="y" /><br />
    합계:
    <input id="sum" /><br />
    <input type="button" value="계산" onclick="calc();" />
  </form>

</body>
</html>
```

웹브라우저에서 실행

덧셈 계산기

첫째 정수: 10

두번째 정수: 20

합계: 30

계산



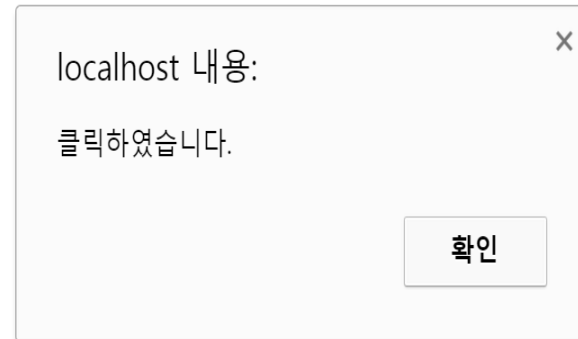
2. UI 구현하기(2001020225_16v4.2)

alert 함수

자바스크립트 다이얼로그 : 경고 다이얼로그

- alert("메시지") 함수
 - 메시지'와 '확인' 버튼을 가진 다이얼로그 출력, 메시지 전달

```
alert("클릭하였습니다.");
```

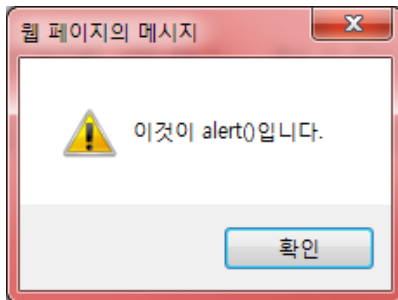




2. UI 구현하기(2001020225_16v4.2)

alert 함수

```
<script>  
    alert("이것이 alert()입니다.");  
</script>
```





2. UI 구현하기(2001020225_16v4.2)

Confirm()함수

❖ Confirm() 함수

- 입력 받을 때 사용
- prompt() 함수와 비슷한 방식으로 사용
 - ✓ 사용자가 확인을 누르면 **true** 리턴
 - ✓ 취소를 누르면 **false** 리턴

코드 :confirm() 함수

```
<script>
  // 변수를 선언합니다.
  var input = confirm("수락하시겠습니까?");

  // 출력합니다.
  alert(input);
</script>
```

그림 :confirm() 함수 사용

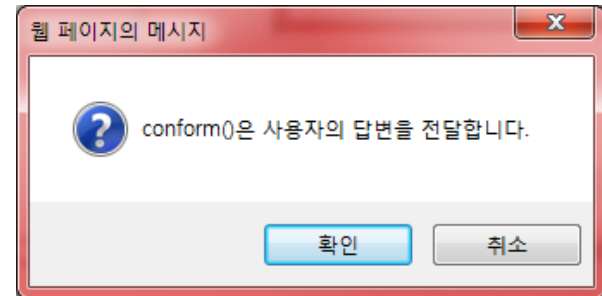




2. UI 구현하기(2001020225_16v4.2)

Confirm()함수

```
<script>  
  var user = confirm("confirm()은 사용자의 답변을 전달합니다.");  
</script>
```





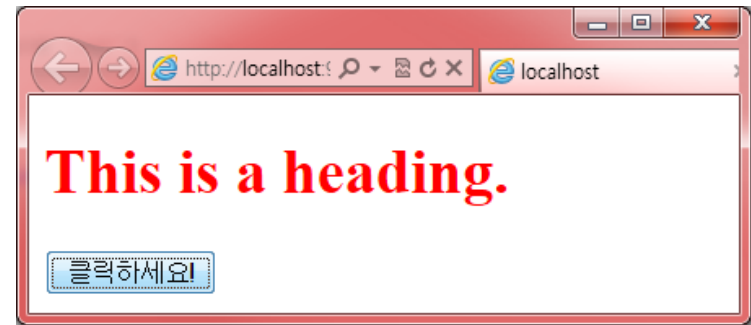
2. UI 구현하기(2001020225_16v4.2)

HTML 요소에 접근하기

```
<!DOCTYPE html>
<html>
<body>

  <h1 id="test">This is a heading.</h1>
  <script>
    function func() {
      e = document.getElementById("test");
      e.style.color = "red";
    }
  </script>
  <button type="button" onclick="func()">클릭하세요!</button>
</body>
</html>
```

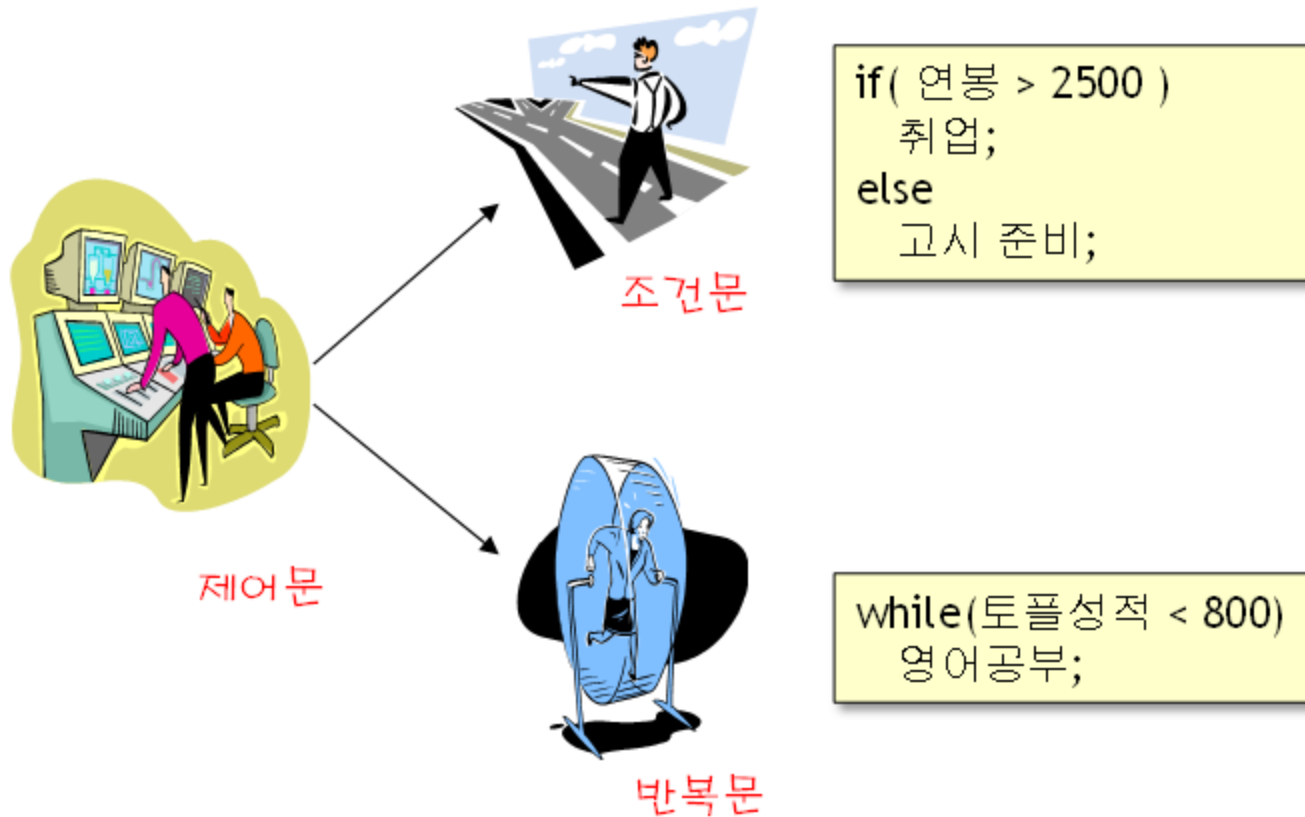
웹브라우저에서 실행





2. UI 구현하기(2001020225_16v4.2)

제어문





2. UI 구현하기(2001020225_16v4.2)

조건문의 종류

- if 문
- if...else 문
- switch 문



2. UI 구현하기(2001020225_16v4.2)

if 문

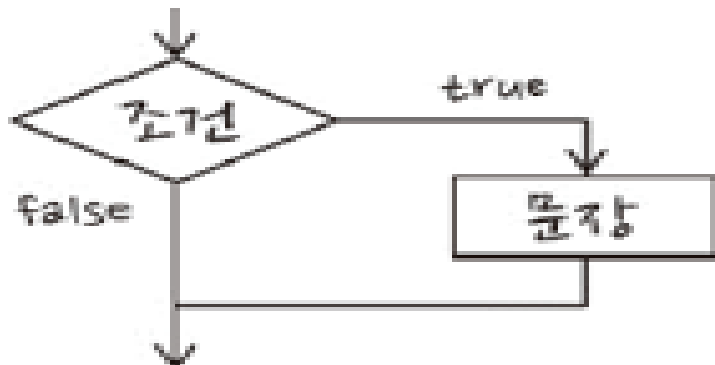
❖ if 조건문

- 자바스크립트에서 가장 일반적인 조건문
 - ✓ 형태

```
if (불 표현식) {  
    문장  
}
```

- ✓ 불리언 표현식이 **true**면 문장 실행
- ✓ **false**면 문장 무시
 - 조건문에 의해 여러 문장을 실행할 때는 중괄호로 감싸야 함

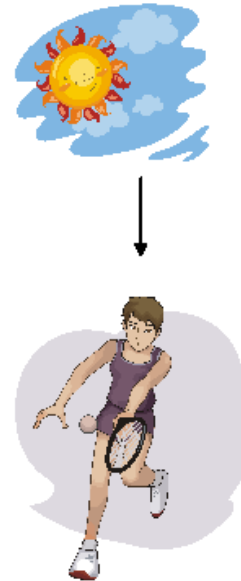
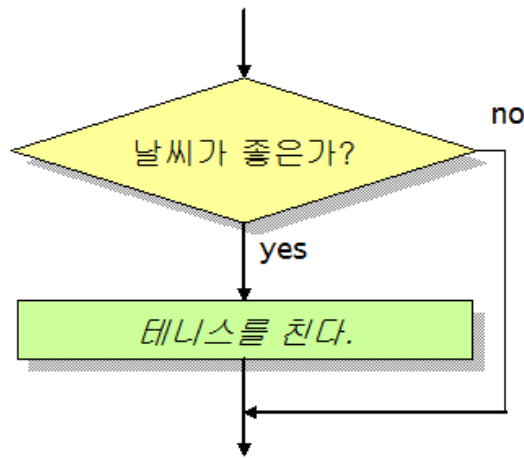
그림 if 조건문





2. UI 구현하기(2001020225_16v4.2)

if 문



```
if (time<12)
{
    greeting="Good Morning!";
}
```



2. UI 구현하기(2001020225_16v4.2)

if 문

❖ 예제 (1)

- 실행 결과 : '종료'가 입력된 경고창만 출력

코드 : if 조건문

```
<script>
  // 조건문
  if (273 < 100) {
    // 표현식 '273 < 100'이 참일 때 실행합니다.
    alert('273 < 100 => true');
  }

  // 프로그램 종료
  alert('종료');
</script>
```



2. UI 구현하기(2001020225_16v4.2)

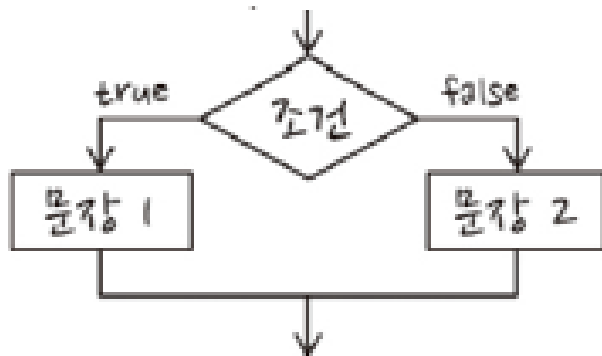
if else 조건문

❖ if else 조건문

- 서로 반대되는 조건에 사용하는 조건문
- **else** 키워드는 **if** 조건문과 함께 사용하는 키워드
 - ✓ if 조건문의 바로 뒤에 붙여 사용
- 조건문의 형태

```
if (불 표현식) {  
    문장 A  
} else {  
    문장 B  
}
```

그림 : if else 조건문





2. UI 구현하기(2001020225_16v4.2)

if-else 문

```
if (time<12)
{
    msg="Good Morning!";
}
else{
    msg="Good Afternoon!";
}
```



2. UI 구현하기(2001020225_16v4.2)

중첩 조건문

❖ 중첩 조건문

- 조건문 안에 조건문을 중첩해 사용하는 형식
- 중첩 조건문의 형태
 - ✓ 여러 번 중첩해도 상관 없음

```
if (불 표현식) {  
    if (불 표현식) {  
        문장  
    } else {  
        문장  
    }  
} else {  
    if (불 표현식) {  
        문장  
    } else {  
        문장  
    }  
}
```



2. UI 구현하기(2001020225_16v4.2)

중첩 조건문

❖ 중첩 조건문

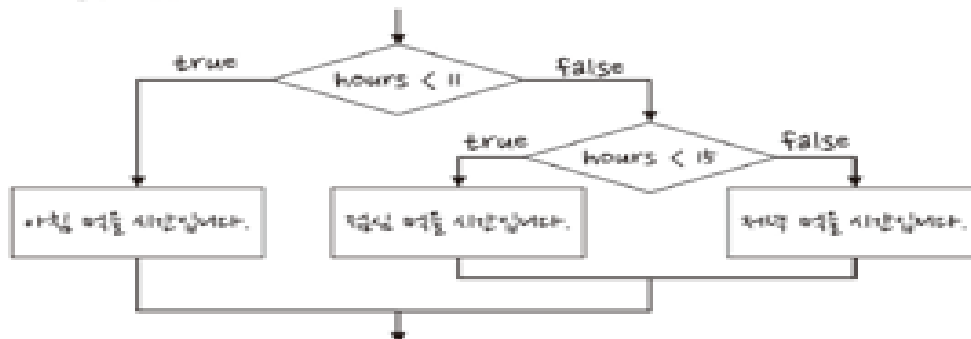
- 진행과정

코드 중첩 조건문

```
<script>
// 변수를 선언합니다.
var date = new Date();
var hour = date.getHours();

// 조건문
if (hour < 11) {
    // 'hour < 11'에 참일 때 실행합니다.
    alert('아침 역을 시간입니다. ');
} else {
    // 'hour < 11'에 거짓일 때 실행합니다.
    if (hour < 15) {
        // 'hour < 15'가 참일 때 실행합니다.
        alert('점심 역을 시간입니다. ');
    } else {
        // 'hour < 15'가 거짓일 때 실행합니다.
        alert('저녁 역을 시간입니다. ');
    }
}
</script>
```

그림 : 중첩 조건문





2. UI 구현하기(2001020225_16v4.2)

if else if 조건문

❖ If else if 조건문

- If else if 조건문의 형태
- 총첩 조건문을 if else if 조건문으로 형태로 변경
→ 한쌍의 중괄호를 삭제

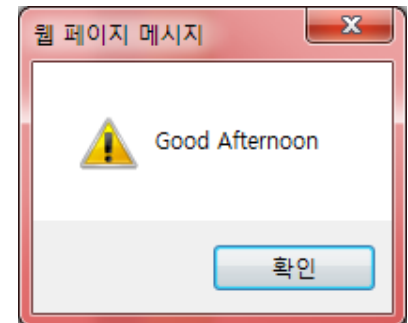
```
if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else if (불 표현식) {  
    문장  
} else {  
    문장  
}
```



2. UI 구현하기(2001020225_16v4.2)

연속적인 if 문

```
<script>
  var msg = "";
  var time = new Date().getHours();
  if (time < 12) {                                // 12시 이전이면
    msg = "Good Morning";
  }
  else if (time < 18) {                           // 오후 6시 이전이면
    msg = "Good Afternoon";
  }
  else {                                           // 그렇지 않으면(오후 6시
    msg = "Good evening";
  }
  alert(msg);
</script>
```





2. UI 구현하기(2001020225_16v4.2)

실습하기

❖ if else if 조건문 실습

1. 사용자에게 숫자를 입력 받아 양수, 0, 음수를 구분하는 프로그램(실습1.html)

실습1

이 페이지 내용:

정수를 입력하시오 :

확인 취소



실습1

10는 양수입니다.

실습1

이 페이지 내용:

정수를 입력하시오 :

확인 취소



실습1

-10는 음수입니다.

실습1

이 페이지 내용:

정수를 입력하시오 :

확인 취소



실습1

0는 0입니다.



2. UI 구현하기(2001020225_16v4.2)

실습하기

❖ if else if 조건문 실습

2. 사용자에게 숫자를 입력 받아 홀수와 짝수를 구분하는 프로그램(실습2.html)

A screenshot of a web browser window with three tabs labeled '실습1'. The address bar shows a file path: 'file:///D:/backup/양정1/2017년%206월%2028일%20...'. The page content displays '이 페이지 내용:' followed by '정수를 입력하시오 :'. Below this is a text input field containing the number '12'. At the bottom right are two buttons: '확인' (Confirm) and '취소' (Cancel).



A screenshot of the same web browser window. The page content now displays '12는 짝수입니다.' (12 is an even number).

A screenshot of the web browser window. The text input field now contains the number '5'.



A screenshot of the same web browser window. The page content now displays '5는 홀수입니다.' (5 is an odd number).



2. UI 구현하기(2001020225_16v4.2)

실습하기

❖ if else if 조건문 실습

3. 사용자에게 국어, 영어, 수학 점수를 입력 받아 평균을 구한 뒤 수우미양가를 구분하는 프로그램(실습3.html)

실습1

이 페이지 내용:

국어 점수를 입력하십시오 :

확인 취소

실습1

이 페이지 내용:

영어 점수를 입력하십시오 :

확인 취소

실습1

이 페이지 내용:

수학 점수를 입력하십시오 :

확인 취소

실습1

국어점수 : 89
영어점수 : 78
수학점수 : 59
평균점수 : 75.33333333333333
성적은 미입니다.



2. UI 구현하기(2001020225_16v4.2)

switch 문

❖ switch 조건문의 기본 형태

- default 부분은 생략 가능

```
switch (비교할 값) {  
    case 값:  
        문장  
        break;  
    case 값:  
        문장  
        break;  
    default:  
        문장  
        break;  
}
```



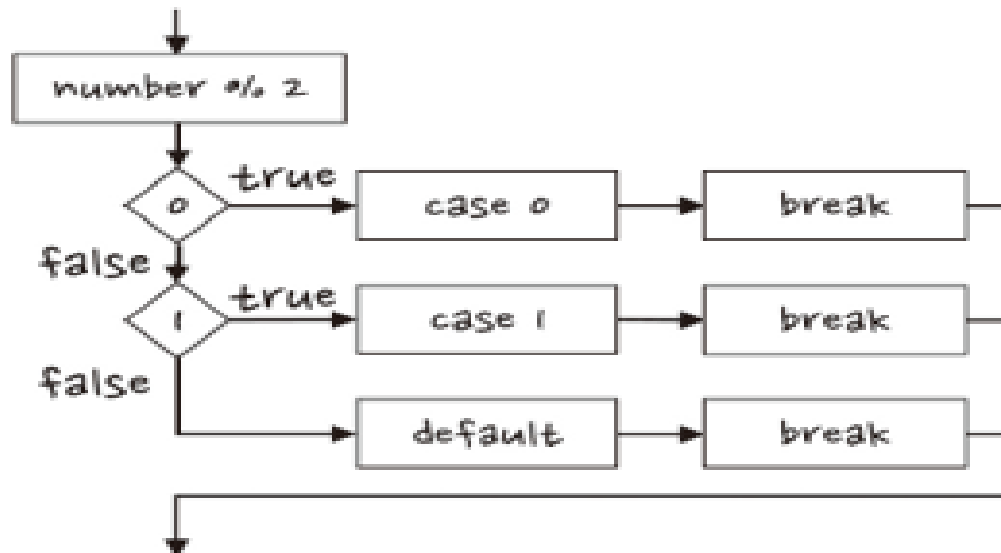
2. UI 구현하기(2001020225_16v4.2)

switch 문

❖ Break 키워드의 의미

- switch 조건문이나 반복문을 빠져 나가려고 사용하는 키워드
- switch 조건문의 괄호 안에는 비교할 값을 입력
 - ✓ 입력한 값을 기준으로 특정 코드 실행
 - ✓ 입력한 표현식과 **case** 키워드 옆의 표현식이 같음
 - **case** 키워드 바로 다음에 오는 문장 실행

그림 : switch 조건문





2. UI 구현하기(2001020225_16v4.2)

switch 문

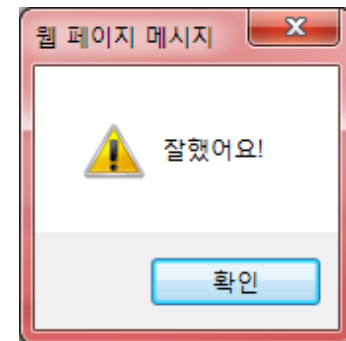
```
<script>
  var grade = prompt("성적을 입력하시오:", "A-F사이의 문자로");
  switch (grade) {
    case 'A': alert("잘했어요!");
              break;
    case 'B': alert("좋은 점수군요");
              break;
    case 'C': alert("괜찮은 점수군요");
              break;
    case 'D': alert("좀더 노력하세요");
              break;
    case 'F': alert("다음학기 수강하세요");
              break;
    default: alert("알수없는 학점입니다.")
  }
</script>
```

localhost에 정보가 필요합니다.

스크립트 프롬프트:
성적을 입력하시오:

A

확인 취소





2. UI 구현하기(2001020225_16v4.2)

숫자 게임 예제

```
<html>
<head>
  <title></title>
  <script>
    var computerNumber = 53;      // 정답
    var nGuesses = 0;             // 추측 횟수

    function guess() {
      var result = "";            // 결과 메시지

      // 사용자가 입력한 값을 받아서 변수 number에 대입한다.
      var number = parseInt(document.getElementById("user").value);
      nGuesses++;                 // 추측 횟수를 증가시킨다.

      if (number == computerNumber) result = "성공입니다.";
      else if (number < computerNumber) result = "낮습니다.";
      else result = "높습니다.";

      document.getElementById("result").value = result;
      document.getElementById("guesses").value = nGuesses;
      return true;
    }
  </script>
</head>
```



2. UI 구현하기(2001020225_16v4.2)

숫자 게임 예제

```
<body>
  <h2>숫자 맞추기 게임</h2>

  이 게임은 컴퓨터가 생성한 숫자를 맞추는 게임입니다. 숫자는 1부터 100 사이에
  있습니다.
  <form>
    숫자:
    <input type="text" id="user" size="5">
    <input type="button" value="확인" onclick="guess();">
    추측횟수:
    <input type="text" id="guesses" size="5">
    힌트:
    <input type="text" id="result" size="16">
  </form>
</body>
</html>
```

The screenshot shows a web browser window with the URL `http://localhost:7424/Projects/HTMLPz`. The page title is "숫자 맞추기 게임". The content includes a description of the game, a form with a text input for the number (containing "23"), a "확인" button, a text input for the number of guesses (containing "1"), and a hint text input (containing "낮습니다").

숫자 맞추기 게임

이 게임은 컴퓨터가 생성한 숫자를 맞추는 게임입니다. 숫자는 1부터 100 사이에 있습니다.

숫자:

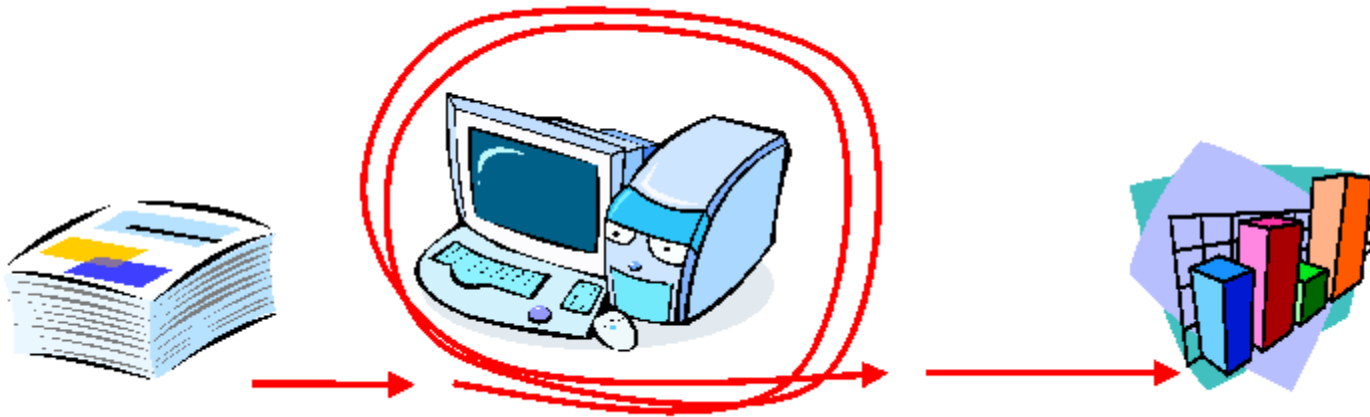
추측횟수: 힌트:



2. UI 구현하기(2001020225_16v4.2)

반복문

- 같은 처리 과정을 여러 번 되풀이하는 것





2. UI 구현하기(2001020225_16v4.2)

반복문

❖ 반복문

- 여러 번 반복해야 하는 일을 간편하게 처리하는 반복문

```
<script>
  alert('출력');
  alert('출력');
  alert('출력');
  alert('출력');
  alert('출력');
</script>
```

```
<script>
  for (var i = 0; i < 100; i++) {
    alert('출력');
  }
</script>
```



2. UI 구현하기(2001020225_16v4.2)

반복문의 종류

- for 문

1 2 4
for(초기문; 조건식; 반복 후 작업) {
.. 작업문 ..
3 }

```
// 0에서 9까지 출력  
for(var i=0; i<10; i++) {  
    document.write(i);  
}
```

0123456789

- while 문

1
while(조건식) {
.. 작업문 ..
2 }

```
var i=0;  
while(i<10) { // i가 0에서 9까지 반복  
    document.write(i);  
    i++;  
}
```

0123456789

- do-while 문

1
do {
.. 작업문 ..
} **while**(조건식);
2

```
var i=0;  
do { // i가 0에서 9까지 반복  
    document.write(i);  
    i++;  
} while(i<10);
```

0123456789



2. UI 구현하기(2001020225_16v4.2)

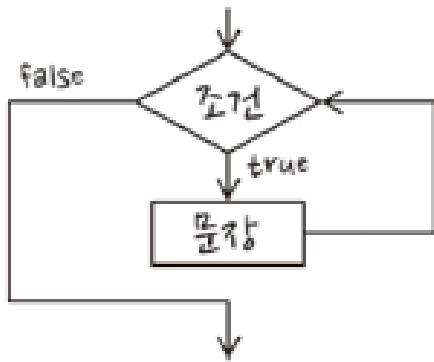
while 문

❖ While 반복문

- ✓ 가장 간단한 반복문
- ✓ if 조건문 형태가 비슷

```
while (불 표현식) {  
    문장  
}
```

그림 while 반복문





2. UI 구현하기(2001020225_16v4.2)

while 문

```
var i = 0;
```

```
while( i < 10 ) ←반복 조건, 변수 i가 10보다 작으면 반복 계속
```

```
{
```

```
    document.write(i+" <br>"); ←반복되는 문장, i를 화면에 출력한다.
```

```
    i++; ←한 번의 반복마다 i를 하나 증가시킨다.
```

```
}
```

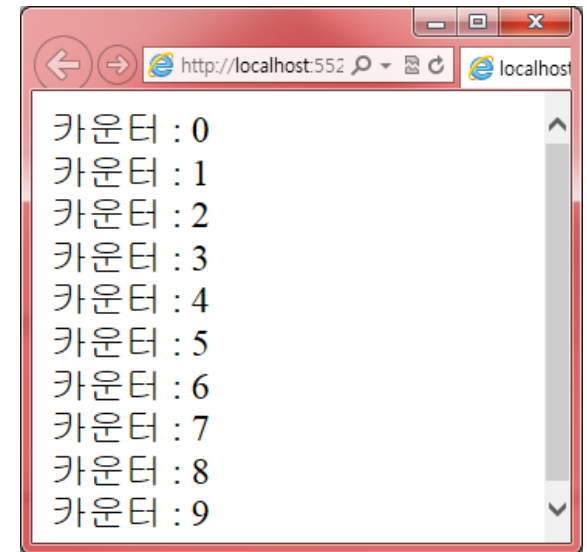
이 문장이 없으면 무한히 반복한다.



2. UI 구현하기(2001020225_16v4.2)

while 문

```
<script>
  var i = 0;
  while (i < 10) {
    document.write("카운터 : " + i + "<br />");
    i++;
  }
</script>
```





2. UI 구현하기(2001020225_16v4.2)

while 문

❖ While 반복문

- ✓ 무한루프 : 무한히 반복문이 반복
- ✓ 익스플로러에서 무한루프의 비극적인 결과로 실행 하지 않도록 주의

```
while (true) {  
    alert('반복문');  
}
```



2. UI 구현하기(2001020225_16v4.2)

while 문

❖ While 반복문

- while 반복문 종료
 - ✓ 숫자 증가
 - ✓ 내부적으로 변화

코드 while 반복문

```
<script>
  // 변수를 선언합니다.
  var value = 0;

  // 반복문
  while (value < 5) {
    alert(value + '번째 반복');
    value++;
  }
</script>
```

- ✓ 시간을 조건으로 변화



2. UI 구현하기(2001020225_16v4.2)

do~while 문

❖ do While 반복문

- 조건의 참 거짓 여부와 상관없이 내부의 문장을 최소한 한 번은 실행해야 하는 경우 사용
- while 반복문과 형태가 비슷
- do while 반복문의 기본 형태

코드 while 반복문

```
<script>
  // 변수를 선언합니다.
  var value = 0;

  // 반복문
  while (value < 5) {
    alert(value + '번째 반복');
    value++;
  }
</script>
```



2. UI 구현하기(2001020225_16v4.2)

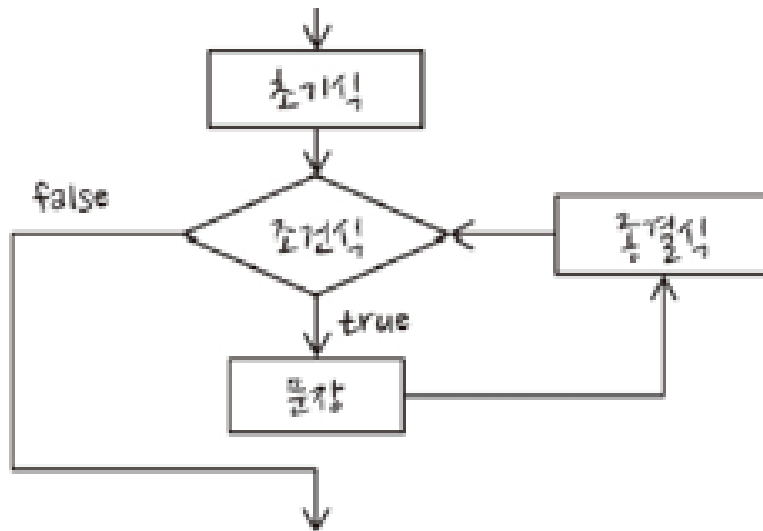
for 문

❖ for 반복문

- 조건(**while** 반복문)보다 횟수에 비중을 둘 때 사용 하는 반복문
- 초기식과 종결식이 있음

```
for (초기식; 조건식; 종결식) {  
    문장  
}
```

그림 for 반복문





2. UI 구현하기(2001020225_16v4.2)

for 문

❖ for 반복문 실행 순서

- ✓ 초기식 실행
- ✓ 조건식 비교/ 조건이 거짓이면 반복문 종료
- ✓ 문장을 실행
- ✓ 종결식 실행
- ✓ 다시 '조건식 비교' 이후를 반복



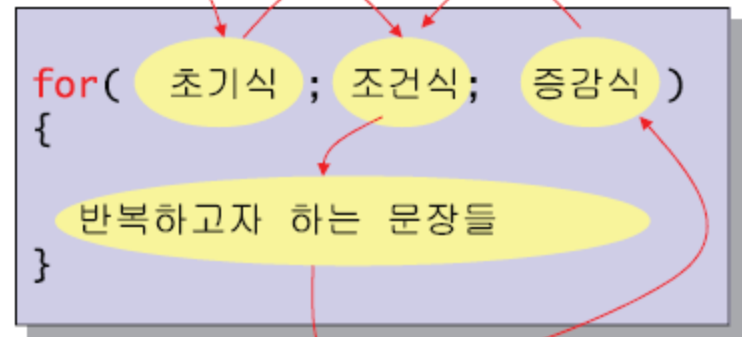
2. UI 구현하기(2001020225_16v4.2)

for 문

```
for( 초기식 ; 조건식 ; 증감식 )  
{  
    document.write(i+"<br />");  
}
```

초기식 조건식 증감식

반복되는 문장

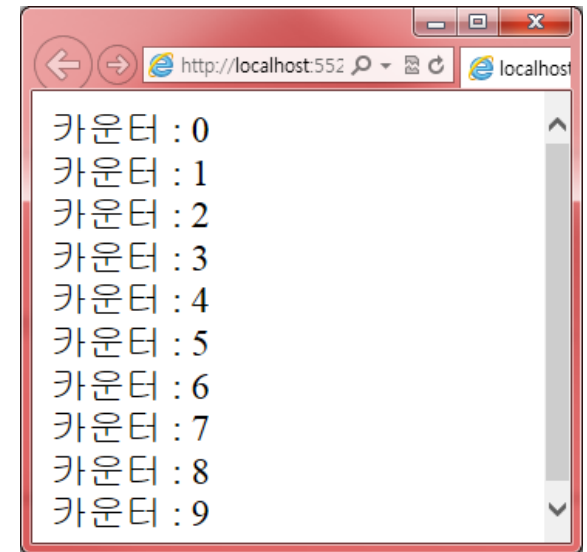




2. UI 구현하기(2001020225_16v4.2)

for 문

```
<script>
  var i = 0;
  for (i = 0; i < 10; i++) {
    document.write("카운터 : " + i + "<br />");
  }
</script>
```



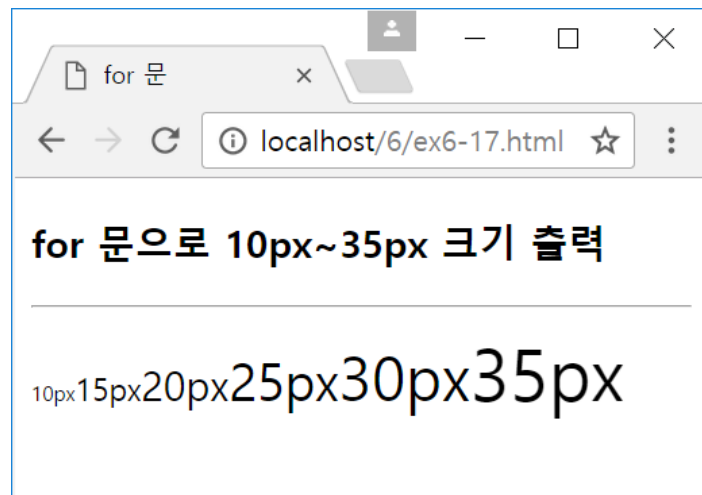


2. UI 구현하기(2001020225_16v4.2)

예제

예제 for 문으로 10px~35px 크기로 출력

```
<!DOCTYPE html>
<html>
<head>
<title>for 문</title>
</head>
<body>
<h3>for 문으로 10px~35px 크기 출력</h3>
<hr>
<script>
  for(var size=10; size<=35; size+=5) { // 5씩 증가
    document.write("<span ">
    document.write("style='font-size:" + size + "px'>");
    document.write(size + "px");
    document.write("</span>");
  }
</script>
</body>
</html>
```





2. UI 구현하기(2001020225_16v4.2)

중첩 반복문 예제

❖ 중첩 반복문

- ✓ 반복문을 여러 겹 중첩해서 사용
- ✓ 중첩 반복문의 예제 (1)

그림 4-8 피라미드(1)



코드 : 만들어보세요

```
<script>
    // 변수를 선언합니다.
    var output = '';

    // 반복문
    // 여기를 작성하세요.

    // 출력합니다.
    alert(output);
</script>
```



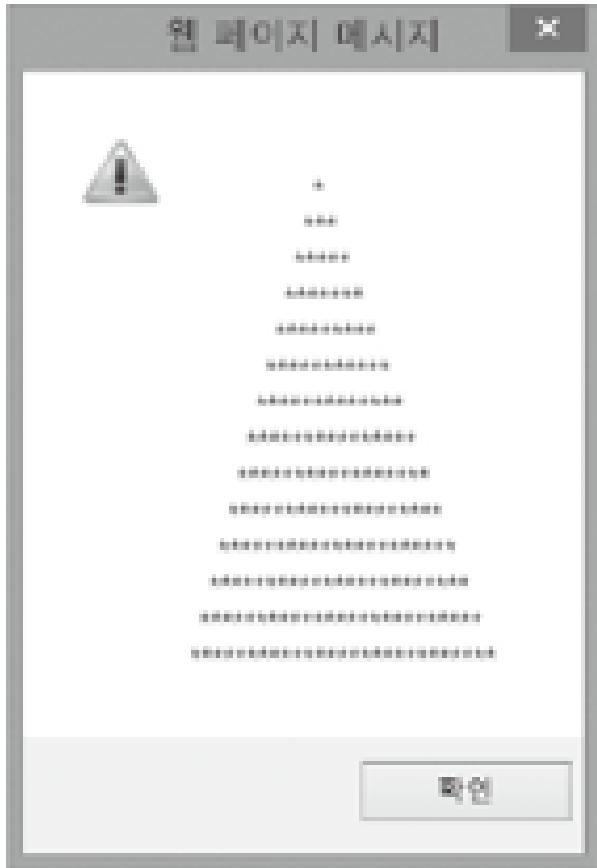
2. UI 구현하기(2001020225_16v4.2)

중첩 반복문 예제

❖ 중첩 반복문

- 중첩 반복문의 예제 (2)

그림 . 피라미드(2)





2. UI 구현하기(2001020225_16v4.2)

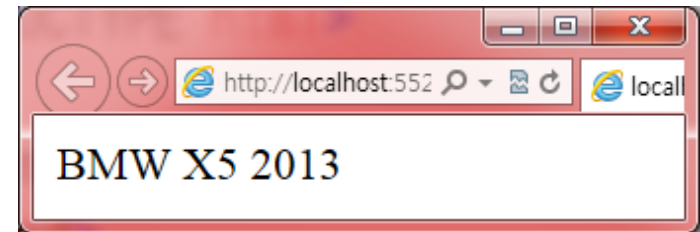
for/in 반복문

```
<script>
  var myCar = { make: "BMW", model: "X5", year: 2013 };
  var txt="";

  for (var x in myCar) {

    txt += myCar[x] + " ";

  }
  document.write(txt);
</script>
```





2. UI 구현하기(2001020225_16v4.2)

break;

❖ break 키워드

- switch 조건문이나 반복문을 벗어날 때 사용
- 다음 반복문은 조건이 항상 참으로 무한 반복
- 무한루프는 **break** 키워드 사용으로 탈출

```
while (true) {  
  
}
```



2. UI 구현하기(2001020225_16v4.2)

continue;

❖ continue 키워드

- 현재 반복을 멈추고 다음 반복을 진행
- continue 키워드 예제 (1)

코드 continue 키워드(1)

```
<script>
    // 반복문

    for (var i = 0; i < 5; i++) {
        // 현재 반복을 중지하고 다음 반복을 수행합니다.
        continue;
        alert(i); // (ㅠㅠ)
    }
</script>
```



2. UI 구현하기(2001020225_16v4.2)

continue;

❖ continue 키워드

- continue 키워드 예제 (2)
- 0부터 10까지의 짝수 합을 구하는 예제
 - ✓ 조건문을 사용해서 홀수일 때는 **continue** 키워드를 만나 바로 다음 반복으로
짝수의 합만 구해짐

코드 | continue 키워드(2)

```
<script>
// 변수를 선언합니다.
var output = 0;

// 반복문
for (var i = 1; i <= 10; i++) {
    // 조건문
    if (i % 2 == 1) {
        // 홀수이면 현재 반복을 중지하고 다음 반복을 수행합니다.
        continue;
    }
    output += i;
}
// 출력합니다.
alert(output);
</script>
```

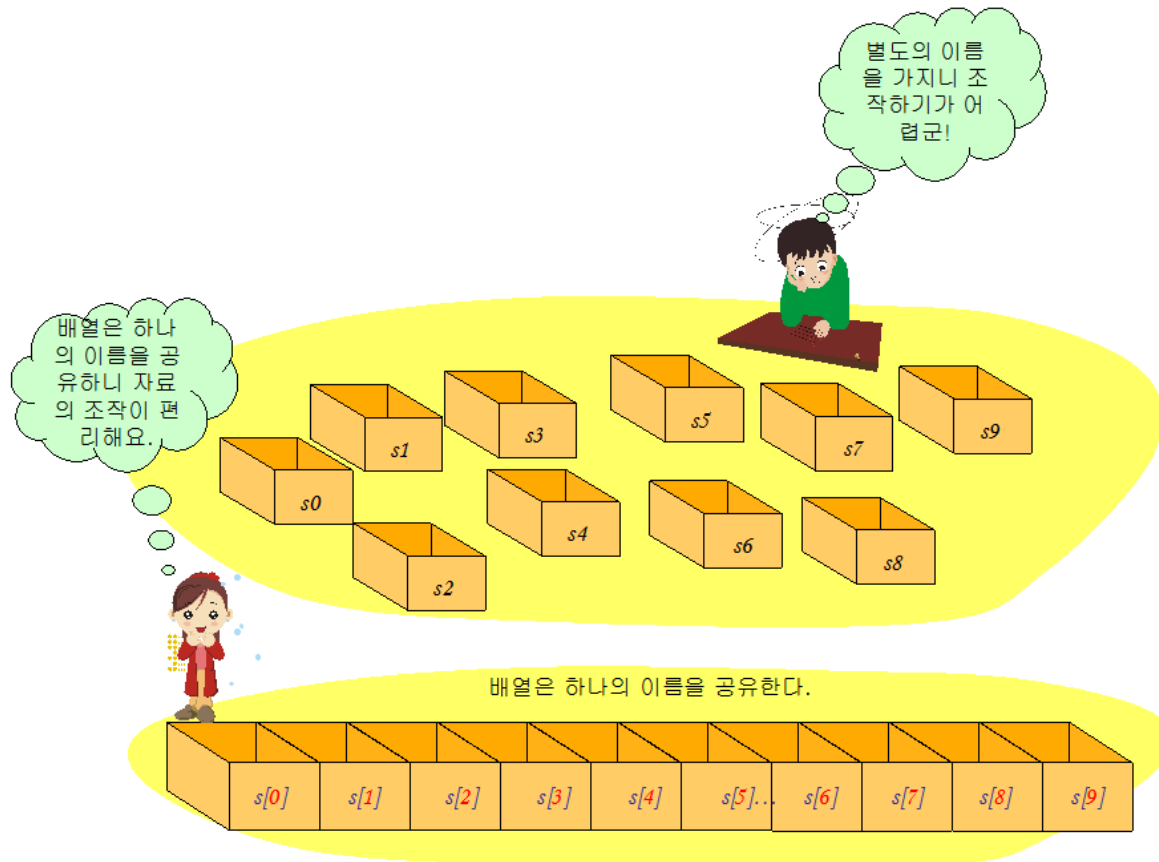
- continue 키워드나 break 키워드의 적당한 사용이 바람직



2. UI 구현하기(2001020225_16v4.2)

배열

- 많은 값을 저장할 수 있는 공간이 필요할 때 배열을 사용한다.
- 서로 관련된 데이터를 차례로 접근하여서 처리할 수 있다.





2. UI 구현하기(2001020225_16v4.2)

배열을 생성하는 2가지 방법

- 리터럴로 배열 생성
 - `var fruits = ["apple", "banana", "peach"];`
- Array 객체로 배열 생성
 - `var fruits=new Array("apple","banana","orange");`



2. UI 구현하기(2001020225_16v4.2)

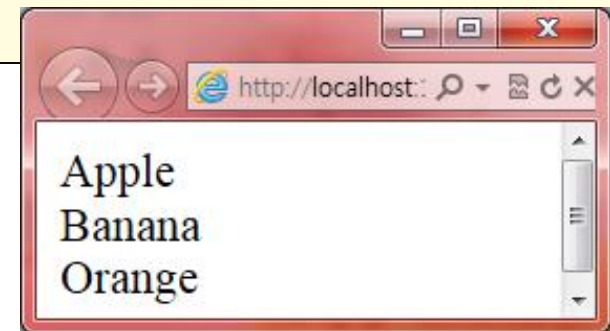
예제

```
<!DOCTYPE html>
<html>
<body>

<script>
  var i;
  var fruits = new Array();
  fruits[0] = "Apple";
  fruits[1] = "Banana";
  fruits[2] = "Orange";

  for (i = 0; i < fruits.length; i++) {
    document.write(fruits[i] + "<br>");
  }
</script>

</body>
</html>
```





2. UI 구현하기(2001020225_16v4.2)

연관 배열

```
<!DOCTYPE html>
<html>
<body>
  <form name="myForm">
    필드1<input type="text" name="a0"><br />
    필드2<input type="text" name="a1"><br />
    필드3<input type="text" name="a2"><br />
    <input type="button" value="초기화" onclick="init();">
  </form>

  <script>
    function init() {
      for (var i = 0; i < 3; i++) {
        document.myForm["a" + i].value = i;
      }
    }

  </script>

</body>
</html>
```

필드1	0
필드2	1
필드3	2

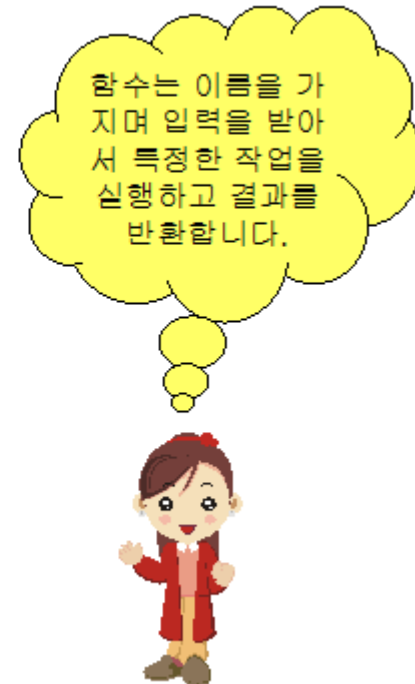
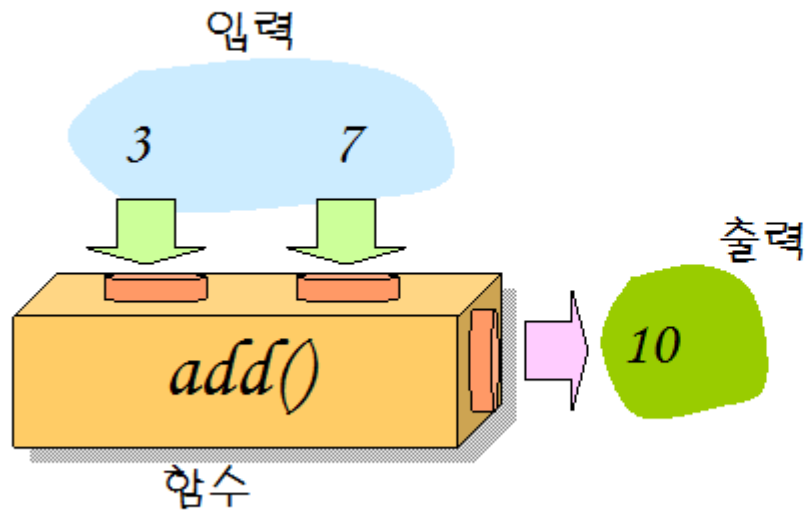
초기화



2. UI 구현하기(2001020225_16v4.2)

함수

- 함수는 입력을 받아서 특정한 작업을 수행하여서 결과를 반환하는 블랙 박스



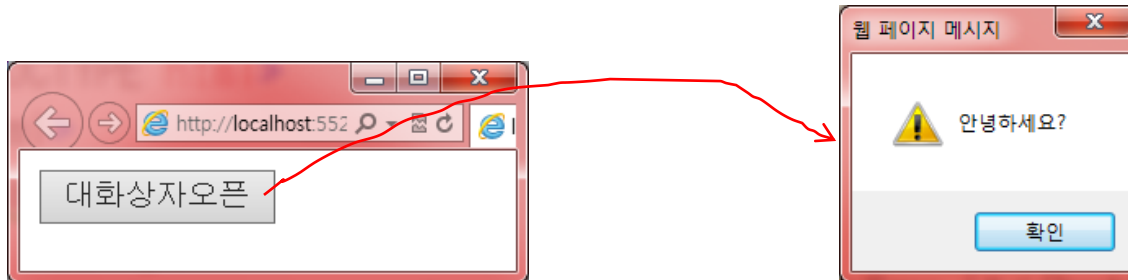


2. UI 구현하기(2001020225_16v4.2)

예제

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function showDialog() {
      alert("안녕하세요?");
    }
  </script>
</head>

<body>
  <button onclick="showDialog()">대화상자오픈</button>
</body>
</html>
```

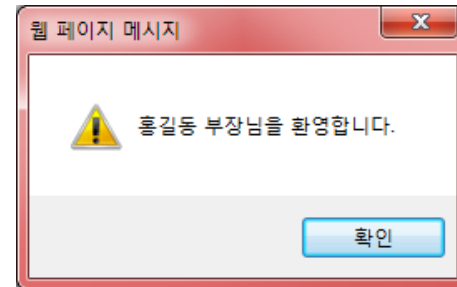
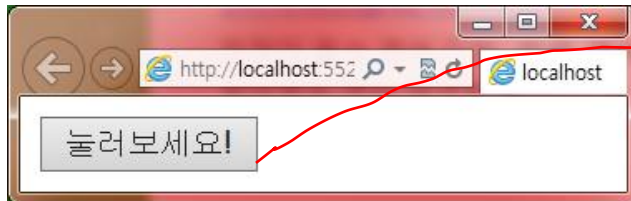




2. UI 구현하기(2001020225_16v4.2)

인수와 매개 변수

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function greeting(name, position) {
      alert(name + " " + position + "님을 환영합니다.");
    }
  </script>
</head>
<body>
  <button onclick="greeting('홍길동', '부장')">눌러보세요!</button>
</body>
</html>
```





2. UI 구현하기(2001020225_16v4.2)

함수의 반환값

- return 문장을 사용하여 외부로 값을 반환

