자바 스크립트

류관희
충북대학교

자바스크립트 – 스크립트의 등장과 역할

- 최초의 스크립트 : 1987년 애플사의HyperCard
- 발전계기 : 1990년대초에 MS사에서 VB에서 사용할 수 있는 VBA(VBApplication) 개발
- SunMicrosystems사가 인터넷 프로그래밍 언어로 Java를 개발
- 넷스케이프사는 선사와 전략적 제휴를 통하여, HTML 기능을 수용하면서 프로그래밍 개념을 대폭 수용한 JavaScript 개발

자바스크립트 – 특징

- 서버가 아닌 클라이언트에서 인터프리터
- 다이나믹 바인딩이 된다.
- 객체 지향형 언어다.
  - 객체 : 윈도, 프레임, URL, 폼, 버튼, 도큐먼트 등
- HTML 문서에 혼합하여 사용한다.
- 변수의 형(type)을 지정할 필요가 없다.
- 일반 유저가 프로그래머 수준의 핸들링을 할 수 있다.

자바스크립트 – 사용 목적

- 인터렉티브(interactive)한 홈페이지
- 경제적인 가격의 컴퓨터로 서버 구축
- 플랫폼이 독립적이다.
- 역동적인 홈페이지 제작
- 웹 프로그램 사용 시 반드시 필요하다.

자바스크립트 – 자바와 자바스크립트

구분	자바	자바스크립트
해석위치	서버	클라이언트
언어형	컴파일러	인터프리터
존재	*.class 파일로 존재	HTML 문서 내에 기술
객체지향	객체 지향 언어	객체 기반 언어
보안성	있음	없음
사용	어려움	쉬움

자바스크립트 – 자바스크립트 기본

기본 사용법
삽입 및 실행법
사용자 정의 함수
작성시 주의사항
특수문자와 주석달기

자바스크립트 – 자바스크립트의 시작

- 자바 스크립트의 삽입 위치는?
- 기본구조

```
<SCRIPT LANGUAGE="JavaScript">
        <!-
            진짜 자바스크립트 코드
        // ->
</SCRIPT>
```

- 자바스크립트의 실행시기는?

자바스크립트 – 자바스크립트 삽입과 실행(1)

- 내장형
- 행 입력형
- 함수형
- 링크형

자바스크립트 – 자바스크립트 삽입과 실행(2)

- 내장형

```
<script language=javascript>
  ~~자바스크립트 소스~~
</script>
```

- 행 입력형

```
<태그 이벤트핸들러="자바스크립트 소스">
```

자바스크립트 – 자바스크립트 삽입과 실행(3)

- 함수형

```
<script language=javascript>
 function 함수명( )
 {
   ~~자바스크립트 소스~~
 }
</script>
<태그명 이벤트핸들러="함수명()">
```

- 링크형

```
<script language=javascript src="js 파일의 전체 경로"></script>
```

자바스크립트 – 주의 사항

- 대소문자를 반드시 구분해야 한다
- 구문은 한 줄에 한 개씩 위치시킨다
- 객체, 속성, 메소드, 함수의 구분은 마침표(.) 연산자를 사용한다.
- 문자열 표시는 따옴표를 사용해야 한다
- 작은따옴표나 큰따옴표를 중첩해서 사용할 때는 반드시 나중에 시작한 따옴표를 먼저 닫아야 한다.
- 따옴표 자체를 문자열에 포함시켜야 할 경우에는 역슬래시(\)와 따옴표를 함께 사용한다.

자바스크립트 – 특수문자

- \n : 개 행(한 줄 바꾸어 출력한다)
- \t : 탭 (일정한 수의 스페이스를 삽입한다)
- \\ : 역슬래시 표시
- \” : 쌍따옴표 표시
- \’ : 온따옴표 표시

자바스크립트 – 주석달기

- 한 행을 주석문 처리

  //주석 처리할 행, 문장
- 두 행 이상에 걸치는 주석문 처리

  /* 주석 처리할 영역 */
- HTML 문서의 주석

  <!– 주석 처리할 영역 -->

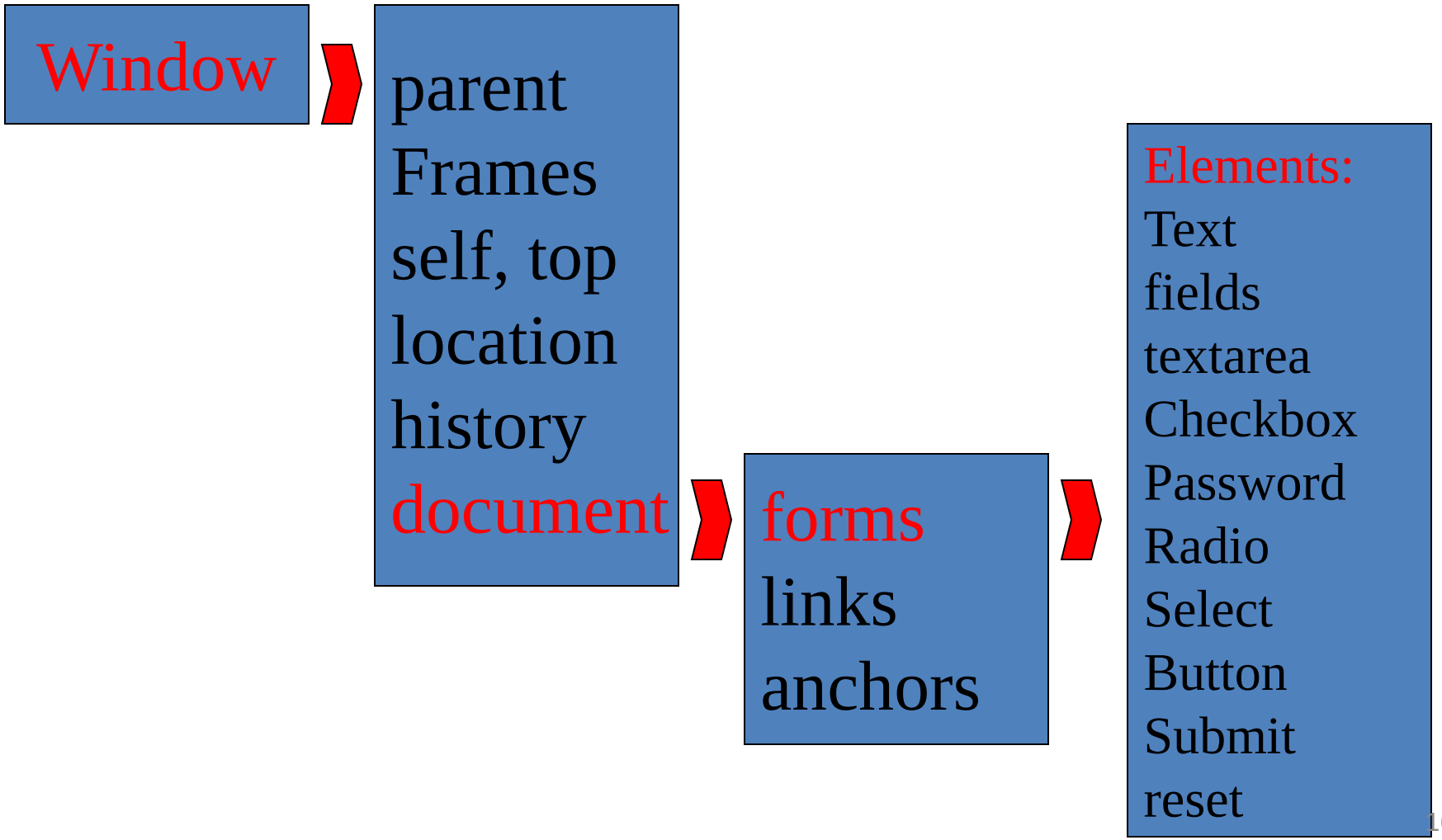
자바스크립트 – 객체

- 객체&속성&메소드 비교
- 객체의 계층 구조
- 객체 표현법
- Window 객체
- Document 객체

자바스크립트 – 객체, 속성, 메소드의 비교

구분	객체	속성	메소드
원어	object	property	method나 function
특징	프로그램의 대상이 되는 모든 것	객체의 속성, 성격, 특징	객체의 기능, 성능, 역할
예	창, 문서	색깔, 크기, 모양	저장, 닫기

자바스크립트 – 객체의 계층 구조



자바스크립트 – 객체 표현법

- 객체명.속성=”값”

  window.status=”GO!”
- 객체명.메소드

  window.close( )
- 상위객체이름.하위객체이름.속성=”값”

  window.document.frm1.id.value=”노무현”

자바스크립트 – **window** 객체

- 윈도의 생성 및 제거 메소드
- open()
  - open(“문서명”, “창이름”, “옵션”)
  - 생성될 윈도의 겉모양
    toolbar,location, status, menubar, scrollbars, resizable, copyhistory, width, height
- close()

자바스크립트 – **document** 객체

- 웹 문서의 색상 설정과 관련된 속성
  fgColor bgColor,alinkColor,linkColor ,vlinkColor ,
- 웹 문서와 관련된 정보를 다루는 속성
  lastModified,location,referrer,title
- 웹 문서에 포함된 내용과 관련된 속성
  anchors,cookie,forms,links
- document 객체의 메소드
  open(),close(),clear(),write(),writeln()

자바스크립트 – 이벤트와 이벤트핸들러

이벤트 & 이벤트핸들러

이벤트핸들러의 종류

자바스크립트 – 이벤트와 이벤트핸들러

- 사용자가 마우스를 움직이거나 키를 누르는 등의 동작을 event라 한다.
- 이 이벤트 앞에 on을 붙이면 event handler가 된다.
- 사용자의 행위 자체는 이벤트
- 사용자의 행위를 전달하는 시점은 이벤트핸들러

자바스크립트 – 이벤트핸들러의 종류(1)

- onLoad( ) : 웹 브라우적 홈페이지를 불러올 때
- onUnload( ) : 현재 홈페이지에서 빠져 나가려할 때
- onClick( ) : 마우스를 클릭할 때

자바스크립트 – 이벤트핸들러의 종류

- onFocus( ) : 양식이나 홈페이지에 커서나 포커스가 위치했을 때
- onBlur( ) : 양식이나 홈페이지에서 커서나 포커스가 다른 곳으로 이동할 때
- onMouseover( ) : 마우스가 위로 왔을 때
- onMouseout( ) : 마우스가 영역을 벗어났을 때

변수, 연산자,
사용자 정의 함수

변수
배열과 객체
연산자
함수

배열, 연산자, 사용자 정의 함수 – 기본 실행문

- 변수 선언문 : var i = 10
- 대입문 : i = 10이나 i = “masan”
- 조건문
  if(i < 10) document.write(“조건만족”)
- 순환문
  for(var i = 0; i < 10; i++){document.write(i)}

배열, 연산자, 사용자 정의 함수 – 변수

- 변수의 데이타 형(type)
  - Numbers(숫자형), String(문자열형)
  - Boolean(논리형), Null(널)
- 변수의 명명시 주의사항
  - 예약어, 함수명, 객체명, 속성명, 사용 중인 변수 등은 사용할 수 없다.
  - 변수는 영자나 밑줄로만 시작한다.
  - 대소문자를 구별하되, 의미있는 이름을 붙인다.

배열, 연산자, 사용자 정의 함수 – 배열 변수 선언법

- 배열(array)은 같은 형, 같은 길이의 데이터를 2개 이상 붙여서 동일한 변수로 처리하는 것
- 기본 형식

  var 배열 변수명 = new Array( )

  배열 변수명[0]=값

  배열 변수명[1]=값

  배열 변수명[2]=값
- var 배열 변수명 = new Array(배열개수)
- var jumsu = new Array(값1, 값2, 값3)

배열, 연산자, 사용자 정의 함수 – 연산문

- 산술연산문 : +, -, *, /, %
  - 증감연산 : ++, --
- 대입연산문 : =, +=, -=, *=, /=, %=
- 조건 연산자:
  - 변수명=(조건식)? 명령1 : 명령2
- 논리연산문 : &&, ||, !,
  - 관계연산자 : >, <, >=, <=
  - 비교연산 : ==, !=
- 연결연산문 : “happy” + “day”

배열, 연산자, 사용자 정의 함수 – 연산기호의 우선순위

- 산술 > 논리 > 대입
  1. ()
  2. ! ++ --
  3. * / %
  4. + -
  5. < <= > >=
  6. == !=
  7. &&
  8. ||
  9. = += -= *= /= %=

배열, 연산자, 사용자 정의 함수 – 사용자 정의 함수

- FUNCTION은 프로그램의 형식을 완전히 갖추지 않은 부속 프로그램으로, 복잡한 계산을 하거나 자주 사용되는 루틴을 정형화할 때 쓰인다.
- 함수의 정의

```
<!-
    function makeWindow(){
      window.open(“allim.htm","new","width=200 height=200")}
//->
```

- 함수의 호출

```
<body onload="makeWindow()">
```

배열, 연산자, 사용자 정의 함수 – 사용자 정의 함수의 종류

- 매개변수가 없는 함수

  Function test(){…}

- 매개변수가 있는 함수

  Function test(name){…}

- 리턴 값이 있는 경우

```
Function test(question){
    Ans=confirm(question)
    Return ans
}
```

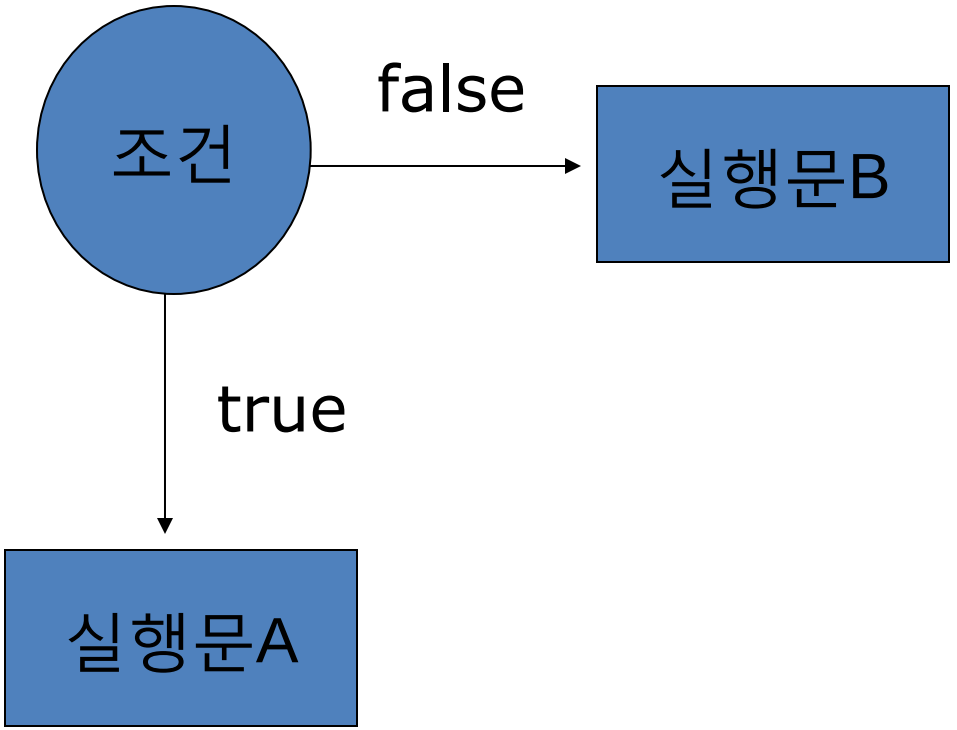
제어문과 내장함수

제어문
내장 함수

제어문
내장 함수

제어문과 내장 함수 – IF 조건문(1)

If(조건)
    실행문A
else
    실행문B



제어문과 내장 함수 – IF 조건문(2)

- 형식1

```
If(조건)
        명령문
```

- 형식2

```
If(조건){
        명령문1
        명령문2
}
else{
        명령문1
        명령문2
}
```

- 중첩 IF문

```
If(조건)
        명령문
else if(조건)
        명령문
else if(조건)
        명령문
else(조건)
        명령문
```

제어문과 내장 함수 – **SWITCH** 문

```
switch(표현식){
    case value1:
            명령문1;
            Break;
    case value2:
            명령문;
            Break;

    ……..
    default
            명령문n
}
```

제어문과 내장 함수 – **FOR** 문 **-** 반복문**(1)**

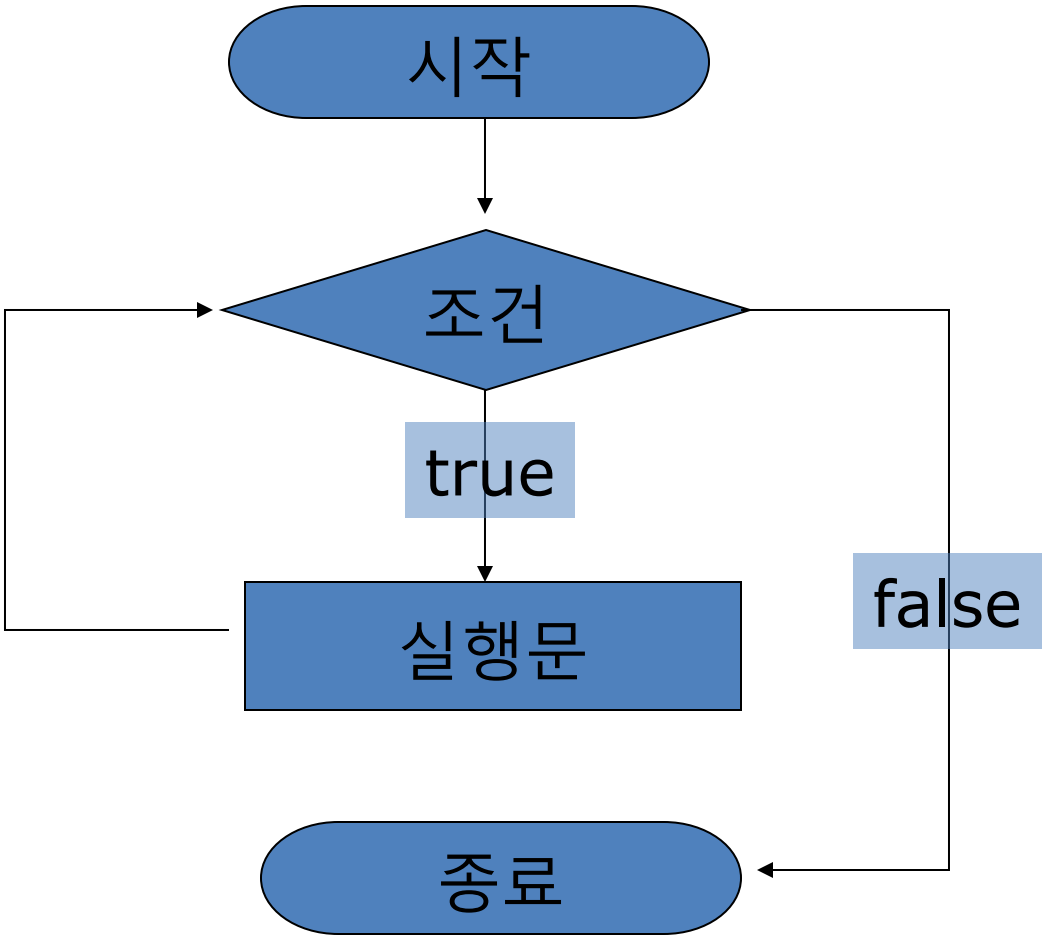
- 기본 형식

```
for(초기 값;조건부;증감식){
    코드부
}
```

- 예제

```
for(a=1;a<11;a++){
    document.write(a+ “*” +a+ “=”+a*a+”<br>”)
}
```

제어문과 내장 함수 – **FOR** 문 **-** 반복문**(2)**



제어문과 내장 함수 – **WHILE** 문

- 기본 형식

```
while(조건){
    명령문
}
```

- 예제

```
a=1
while(a<11){
    document.write(a+ “*” +a+ “=”+a*a+”<br>”)
    A++
}
```

제어문과 내장 함수 – **DO WHILE** 문

- 기본 형식

```
do{
    명령문
} While(조건)
```

- 예제

```
a=1
do{
    document.write(a+ “*” +a+ “=”+a*a+”<br>”)
    A++
} while(a<11)
```

제어문과 내장 함수 – **BREAK, CONTINUE** 문

- Break : 제어문 종료
- Continue : 제어문 반복
- 예제

```
A=10
While(true){
    a—
    if(a>10) continue
    if(a==0) break
    document.write(a+ “*” +a+ “=”+a*a+”<br>”)
}
```

제어문과 내장 함수 – **RETURN** 문

- 함수에서 특정 값을 리턴 값으로 보내고 싶을 때 사용
- 예제

```
Function square(a){
      return a*a
}
--------
For(a=1;a<11;a++){
      document.write(a+ “*” +a+ “=”+a*a+”<br>”)
}
```

제어문과 내장 함수 – **FOR IN**– 객체 조작문

- 객체가 가지는 속성 정보를 알려준다.
- 만약 객체의 모든 속성이 5개라면 5번 반복된다.
- 자바 스크립트는 완성된 언어가 아니므로 버전업 되면서 새로 추가된 객체의 속성 정보를 알 수 있다.
- for (variable in 객체)
  {
    수행할 작업
  }

제어문과 내장 함수 – **WITH** – 객체 조작문

- 하나의 객체에 대해 여러 가지 속성들을 한꺼번에 조작할 때 사용한다.
- with (객체)
  {
    조작 내용
  }
- with (document)
  {
    bgcolor = "white";
    fgcolor = "red";
  }

제어문과 내장 함수 – 내장 함수의 종류

표현식	설명
alert("메시지")	'확인' 버튼이 있는 메시지 창을 띄움
prompt("메시지","기본 문구")	입력상자가 있는 메시지 창을 띄움
confirm("메시지")	'확인', '취소' 버튼이 있는 메시지 창을 띄움
eval( )	문자열을 수식으로 바꿈
isNan( )	전달받은 값이 숫자인지 문자인지 판별하여 숫자가 아닌 경우 true 값을 반환
parseFloart( )	문자열을 부동소수점으로 바꿈
parseInt( )	문자열을 정수로 바꿈
escape( )	ISO-Latin-1 문자 셋을 아스키 값으로 바꿈, 문자 값을 URL 표기형으로 변환
unescape( )	아스키 값을 ISO-Latin-1 문자 셋으로 바꿈, URL 표기형을 문자로 변환
isFinite( )	전달받은 값이 유리수인지 판단하여 유리수인 경우에만 true 값 반환
Number( )	객체를 수치로 변환
String( )	객체를 문자열로 변환