*** overview ***

1. 자바 스크립트란?

- 동적 컴퓨터 프로그래밍 언어
- 웹페이지의 일부로 사용: 클라이언트측 스크립트가 사용자와 상호작용하는 것을 허용해 동적 페이지를 만듬.
- 객체 기능을 가진 인터프리터 프로그래밍 언어
- 유래: LiveScript->JavaScript(NETSCAPE)

2. 클라이언트측 자바스크립트

- 브라우저가 해석/실행: HTML문서에 포함되거나 참조(외부)되어야 함
- 사용자와 상호작용, 브라우저 제어, 동적으로 HTML컨텐츠를 생성
- 유리한점 예: 폼필드내에 사용자가 유효한 메일주소를 입력했는지를 체크
- 사용자가 폼 보내기할 때, 모든 입력이 유효하면 웹브라우저에게 보내짐
- 버튼 클릭 등 사용자가 명시적/암시적으로 생성한 이벤트를 가로챌 수 있음

3. 한계

- 완전한 독립적 프로그래밍 언어가 아님: 기능 부족

4. 목록

25. date

26. math

27. regexp

28. dom

30. formvalidation

35. browsers

*** syntax ***

1. 기본 문장: 웹 페이지의 HTML 태그 안

<script ...>

JavaScript code

</script>

2. 중요한 두 개 속성

가. 언어지정

나. 타입

다. 예

<script language="자바스크립트" type="텍스트파일/자바스크립트">

JavaScript code

</script>

3. 공백, 탭, 뉴라인 무시

4. 세미콜론은 옵션

5. 대소문자 구분

6. 주석문: <!-- //-->

스타일: /* */

HTML: <!-- -->

*** placement ***

1. 위치

- 헤드섹션: <head>...</head>
- 바디섹션: <body>...</body>
- 외부파일: <head>...</head>에서 포함

2. 위치별 차이

- 헤더섹션: 사용자 이벤트 발생할 때

- 바디섹션: 페이지가 로딩될 때(웹페이지내 컨텐츠를 생성/삽입)
- 외부파일:

<script TYPE="타입" SRC="파일경로" ></script>

*** variables ***

1. 데이터 타입

- 숫자: 123, 120.50 주의: 정수/실수 구분X, 64비트 부동소수점
- 문자열: '문자열'
- 논리: true/false
- 특이변수

가. null

나. undefined

다. object

2. 변수: untyped 언어

- var meney;
- money;
- 초기화: 초기화 값에 따라 타입이 정의

3. 변수영역

- 글로벌 변수
- 로컬변수: {}내에 선언

```
myVar = "전역변수";
anyVar = "임의문자";
function checkscope() {
        myVar = "지역변수";
        alert(myVar);
        alert(anyVar);
}
```

4. 변수명

- 예약어 사용금지: if else null switch ...
- 숫자로 시작 x
- 대소문자 구분

*** operators ***

1. 연산자의 종류

- 산술연산
- 비교연산
- 논리연산

- 할당연산
- 조건연산

2. 산술연산
- 일반 사칙연산: +, -, *, / -> 4/2 = 2.0
- 나머지 연산: % -> 4%2 = 0
- 증감연산자: ++, -- -> a = 10; a++; a = a + 2;

3. 비교연산: ==, !=, >, <, >=, <=

4. 논리연산: &&(AND), ||(OR), !(NOT) -> true->false

5. 할당연산: =, +=, -=, *=, /=

6. 기타연산
가. 조건연산: ? -> ((a > b) ? 100 : 200)
나. typeof(단항연산자)
var b = "String"
typeof b -> "string"

*** ifelse ***

- if 문장
- if else 문장
- if else if 문장

2. if 문장
```
if (식){
        참일 경우 실행될 문장
}
```

3. if else 문장
```
if (식){
        참일 경우 실행될 문장
} else {
        거짓일 경우 실행될 문장
}
```

4. if else if 문장
```
if (식1){
        식1 참일 경우 실행될 문장
} else if (식2){
        식2 참일 경우 실행될 문장
```

```
} else if (식3){
        식3 참일 경우 실행될 문장
} else{
        어떠한 조건식도 만족하지 않을 경우 실행될 문장
```

*** functions ***

1. 함수란

- 프로그램안의 임의의 위치에서 호출될 수 있는 재사용가능한 코드 집합
- 코드의 재작성을 필요성을 줄임
- 개발자가 모듈러(규격화 된 부품) 코드를 작성하는데 도움을 줌.
- 개발자가 큰 프로그램을 보다 작은 관리가능한 기능들로 나뉘도록 함.
- 예: alert(), write()

2. 함수정의

```
<script type="타입">
<!--
function 함수이름(파라메터 리스트)
{
        문장;
}
//-->
</script>
```

3. 함수 사용 예: alert(), write()

```
<script type="TEXT/JAVASCRIPT">
<!--
function SAYHELLO()
{
        alert("Hello there");
}
//-->
</script>

<HTML>
<head>
<script type="타입">
function SAYHELLO()
{
        document.write ("Hello there!");
}
</script>
```

```
</head>
<BODY>
<p>Click the following button to call the function</p>
<form>
<input type="button" 클릭이벤트="SAYHELLO()" value="Say Hello">
</form>
<p>Use different text in write method and then try...</p>
</BODY>
</HTML>
```

4. 함수 파라메터: 함수내에서 사용되며, 콤마(,)로 분리되어 있음

5. 함수결과 리턴
- return 문장: 옵션
- 함수의 마지막 문장

6. 함수 중첩
- 자바스크립트 1.2 이상
- 다른 함수안에서 함수정의 가능
- 함수 리터럴

7. 함수생성자(Function() 생성자)
- new 연산자와 함께 사용해 동적으로 함수를 정의
- 문법

```
<script type="TEXT/JAVASCRIPT">
        <!--
                변수명 = new Function(Arg1, Arg2..., "함수몸체");
        //-->
</script>
```

8. 함수 리터럴
- 자바스크립트 1.2 도입: 함수를 정의하는 방법
- 이름없는 함수를 정의하는 표현식
- 문법

```
<script type="타입">
<!--
        변수명 = function(Argument List){
                Function Body
        };
//-->
</script>
```

*** events ***

1. 이벤트
- 사용자가 브라우저가 웹 페이지를 조작할 때 발생
- 웹페이지가 로드될 때, 사용자가 버튼을 클릭할 때
- 키 입력, 윈도우 종료, 윈도우 사이즈 변경 등
- 자바스크립로 코드된 응답을 실행
- DOM(Document Object Model) 레벨3의 부분으로
  모든 HTML 엘리먼트가 자바 스크립트 코드를 일으킬 수 있는 이벤트 집합을 포함
- 참고

가. DOM레벨1은 HTML, XML문서 구조를 정의

나. 레벨2와 레벨3은 레벨1에 따른 상호작용 기능 추가

2. 이벤트 종류
- <body> 레벨: onload/onunload
- <form> 레벨

가. onchange: 엘리먼트가 변할 때

나. onsubmit: 폼이 보내질 때

다. onreset: 폼이 리셋될 때

라. onselect: 엘리먼트가 선택될 때

마. onblur: 엘리먼트가 포커스를 잃을 때

바. onfocus: 엘리먼크가 포커스를 얻을 때
- 키보드: onkeydown/onkeypress(눌렀다 뗄 때)/onkeyup
- 기타: onclick/ondblclick 등등

*** cookies ***

1. 쿠키
- 웹브라우저-웹서버간 통신을 위해 HTTP(TCP/IP기반프로토콜 응용) 프로토콜 사용
- HTTP 프로토콜은 무상태 프로토콜
- 문제: 상업사이트(장보기): 여러 다른 페이지들 사이에 연결(session) 정보가 필요
- 선호도, 구입정보 트래캥의 해결책

2. 작동원리
- 웹서버는 방문자 브라우저에 쿠키형태의 약간의 데이터를 전송
- 웹 브라우저는 쿠키를 받아서 저장
- 저장방식: 방문자 하드디스크에 평문레코드로 저장됨
- 방문자가 사이트의 다른 웹페이지를 가면 웹브라우저는 동일한 쿠키를 웹서버에 전송
- 웹서버는 이전에 저장된 것을 알고 기억함
- 저장형태: 이름=값 형식

3. 자바 스크립트
- 도큐먼트 오브젝트의 쿠기 속성으로 접근(Document.cookie)

- 쿠키저장: document.cookie = "키1=값1;키2=값2";
- 쿠키읽기: split
- 만료설정: expires
- 쿠키삭제: expires

4. 쿠키 실습

페이지1

아이디 입력박스

패스워드 입력박스

보내기 버튼 -> onclick이벤트 처리 checkAuth();

링크생성

링크없음

checkLinkEnable() {

초기화: 아이디

쿠키 체크

쿠키 존재하면

값을 가지고 와서

아이디를 비교

일치 링크추가: 글자

}

checkAuth() {

아이디/패스워드 특정값으로 초기화

입력박스에서 넘어온 값과 비교

일치: 쿠키 생성

불일치: 쿠키 생성

메시지: alert('인증정보 불일치') return;

document.getElementById('').innerHTML =

}

페이지2

*** dialogbox ***

1. 경고 다이얼로그 박스
2. 확인 다이얼로그 박스
3. 프롬프트 다이얼로그 박스

*** Objects ***

1. 객체 지향 언어의 조건
- 캡슐화(Encapsulation): 메소드와 관련정보(데이터)를 함께 저장
- 상속(Inheritance)
- 다형성(POLYMORPHISM): 자식객체를 부모객체가 참조
- 집합성(Aggregation): 한 객체가 다른 객체를 포함

2. 객체구성: 메소드(function==method) + 속성(properties==property)

3. 객체 프로퍼티
- 세 가지 프리미티브 데이터 타입이나 임의의 추상데이터타입
- 객체 메소드 안에서 내부적으로 사용되는 변수
- 웹페이지 전체에서 사용되는 글로벌 변수도 가능
- 형식: 객체이름.프로퍼티=프로퍼티 값;
- 예: var str = document.title;

4. 객체 메소드
- 객체 함수
- 함수와의 차이: 독립적 or 객체 종속
- 예: document.write("이것을 테스트입니다.");

5. 객체의 종류: 내장객체, 사용자 정의 객체

6. 객체 생성과 this
- 객체 생성: new 연산자
- this: 생성자 함수에 넘겨지는 객체 참조자

7. 오브젝트 메소드 정의: 자바스크립트에서는 메소드도 변수다

8. with 키워드
- 객체 메소드나 프로퍼티의 참조 약칭
- 예: with (객체){
        객체 이름 없이 사용된 프로퍼티;
}

*** number ***

1. Number 객체
- 브라우저에 의해 자동변환: var num = 1;
- 문법: var val = new Number(number);
- 숫자 외 값을 사용하면 NaN(Not a number) 리턴

2. Number 프로퍼티
- MAX_VALUE: var val = Number.MAX_VALUE;
- MIN_VALUE
- NaN: 유효한 숫자를 리턴해야 하는 함수의 에러 조건을 체크하는데 사용
- 중략
- prototype: 현재 문서내 새로운 프로퍼티나 메소드 할당
- constructor: 오브젝트 생성 함수 리턴

3. Prototype
- 현재 문서내 임의의 객체에 새로운 프로퍼티나 메소드 할당
- 임의의 객체: Number, Boolean, String, Date 등등
- 문법: 객체명.prototype.name = value

4. constructor

5. Number 메소드
- toString(): number.toString([radix])
radix(진수): 2~36
- valueOf(): 오브젝트의 프리미티브 값, number.valueOf()

*** boolean ***

1. Boolean
- 표현값: true or false
- 기본 false 값: 0, -0, null, false, NaN(Not a Number), undefined

2. 프로퍼티: constructor, prototype

3. 메소드
- toString()
- valueOf()

*** string ***

1. 문법: val = new String(string);

2. 속성
- constructor
- length
- prototype

3. 주요함수

- charAt(): 특정 인덱스의 문자를 리턴
- concat(): 문자열 합치기
- indexOf(): 특정값의 첫번째 발생 위치(없으면: -1)
- replace(): 매칭 문자열을 변경
- search(): 매칭 문자열 찾기 (없으면: -1)
- split(): 특정구분자로 특정갯수로 문자열 분리
- substr(): 시작위치부터 특정길이까지 문자열을 얻어옴(0~길이-1)
- substring(): 시작위치에서 종료위치(옵션)까지 읽어옴
- toLocaleLowerCase(): 소문자로
- toString(): 지정된 객체를 문자열로 리턴
- valueOf(): 문자열 객체의 프리미티브 값을 리턴
- anchor(): 문자열의 HTML래퍼클래스

*** array ***

1. 문법
- fruits = new Array("APPLE", "ORANGE", "MANGO");
- fruits = ["APPLE", "ORANGE", "MANGO"];
- 내용

fruits[0]: 첫 번째 요소

fruits[1]: 두 번째 요소

fruits[2]: 세 번째 요소

2. 속성
- constructor
- length: 길이
- prototype

3. 주요함수
- concat()

*** regexp ***

1. 정규표현식(Regular Expression)
- 캐릭터의 패턴을 서술하는 객체
- 텍스트에 대한 강력한 패턴 매핑과 검색대체 함수 있음
- 문법

가. pattern = new RegExp(pattern, attributes);

나. pattern = /pattern/attributes;

다. pattern: 정규포현식의 패턴을 지정하는 문자열

라. attributes: 옵션(대소문자무시 'i', 매칭전체 'g', 매칭 'm')

2. 표현식과 설명

- [...]: 임의의 하나의 문자
- [^..]: 임의의 하나의 문자가 아닌것
- [0-9]: 0에서 9사이의 문자
- [A-Z]: 영문 대문자
- [a-z]: 영문 소문자
- [a-Z]: 영문 대소문자

3. 퀀티파이어
- p+: 문자 하나이상
- p*: 문자 0개 이상
- p$: 끝나는 문자
- ^p: 시작문자

6. 주요함수
- exec()
- test()

*** forminput ***

<input name="searchTx1" type="text" id="searchTxt1" class="searchField" value="김형태1" />

<input name="searchTx2" type="text" id="searchTxt2" class="searchField" value="김형태2" />

방법 1: 일반적인 text
document.getElementById('searchTxt').value

방법 2:
document.getElementsByClassName('searchField')[0].value -> 김형태1
document.getElementsByClassName('searchField')[1].value -> 김형태2

방법 3:
document.getElementsByTagName('input')[0].value -> 김형태1
document.getElementsByTagName('input')[1].value -> 김형태1

방법 4: 라디오버튼/체크박스
document.getElementsByName('searchTx1')[0].value -> 김형태1
document.getElementsByName('searchTx2')[1].value -> 김형태2

방법 5: CSS 선택자
document.querySelector('#searchTxt1').value

방법 6:
document.querySelectorAll('.searchField')[0].value