

# 자바스크립트

## 1. 에디트 플러스 템플릿 파일 수정하기

- ① 에디트 플러스 실행하기- 도구 메뉴 - 기본 설정
- ② 일반 탭 - XHTML 태그 사용 체크
- ③ 파일 탭 - 기본 인코딩(UTF-8) . 저장시 백업 파일 생성 체크 해제
- ④ 파일 탭 - 문서 템플릿 - XHTML - [열기]
- ⑤ XHTML 기본 문서를 수정후 [저장]합니다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<title> new document </title>
<meta name="generator" content="editplus" />
<meta name="author" content="" />
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

- ⑥ 도구 메뉴 - 기본 설정 - 도구 탭 - Browser 1 : 외부브라우저 - FireFox 선택  
Browser 2 : 외부브라우저 - 인터넷익스플로러 선택

## 2. 자바스크립트 기초 문법

```
<script type="text/javascript">
자바스크립트 실행문;
</script>
```

```
<script type="text/javascript">
// 자바스크립트 주석처리 한 줄 주석
/* 여러 줄 주석처리 */
</script>
```

### (1) 자바스크립트 변수

var 변수명; 또는 var 변수명=값;

```
<script type="text/javascript">
var box;
box=100; //숫자형 데이터
box=30;
document.write(box);
</script>
```

```
<script type="text/javascript">
var s="javascript"; //문자형 데이터
var num="100";
var tag="<h1> String </h1>";
</script>
```

```
var s=true; //논리형 데이터
var t=10>=100;
var k=Boolean("hello");
```

## (2) 자바스크립트 연산자

① 산술 연산자 : 더하기(+), 빼기(-), 곱하기(\*), 나누기(/), 나머지(%)

② 대입 연산자

<b>A = B</b>	→	<b>A = B</b>
<b>A += B</b>	→	<b>A = A + B</b>
<b>A *= B</b>	→	<b>A = A * B</b>
<b>A %= B</b>	→	<b>A = A % B</b>
<b>A -= B</b>	→	<b>A = A - B</b>
<b>A /= B</b>	→	<b>A = A / B</b>

③ 증감 연산자

변수 ++; ex) var a=10; a++; → a=11  
변수 --; ex) var a=10; a--; → a=9

var A=++B // 변수 B 데이터 1 증가후 변수 A에 저장  
var A=B++ // 변수A에 변수B의 데이터 저장 후  
변수 B의 데이터를 1 증가 시킨다.

④ 비교 연산자

<b>A &gt; B</b>	A가 B보다 크다
<b>A &lt; B</b>	A가 B보다 작다
<b>A &gt;= B</b>	A가 B보다 크거나 같다
<b>A &lt;= B</b>	A가 B보다 작거나 같다
<b>A == B</b>	A와 B는 같다
<b>A != B</b>	A와 B는 같지 않다
<b>A === B</b>	A와 B는 데이터 타입과 값이 모두 같다
<b>A !== B</b>	A와 B는 데이터 타입또는 값이 다르다.

⑤ 논리 연산자

<b>  </b>	OR연산자라 부르며 피연산자 중 한 개라도 true면 결과는 true
<b>&amp;&amp;</b>	AND연산자라 부르며 피연산자 중 한 개라도 false면 결과는 false
<b>!</b>	NOT연산자라 부르며 피연산자가 false면 결과는 true

※ 연산자 우선 순위

() > 단항연산자 > 산술연산자 > 비교연산자 > 논리연산자 > 대입(복합대입)연산자

⑥ 삼항 조건 연산자

**조건식 ? 실행문 1 : 실행문 2;** 조건식이 참이면 실행문1을, 거짓이면 실행문2 실행

```
<script type="text/javascript">
  var a=10; var b=3;
  var result= a>b ? "javascript" : "hello";
  document.write(result);
</script>
```

### 3. 제어문

제어문은 프로그램의 흐름을 제어할 수 있도록 도와주는 실행문을 말합니다.

① 조건문 : if 문은 조건식을 만족(true)할 경우에만 실행문을 실행합니다

```
if(조건식) {  
    실행문;  
}
```

```
if(조건식) {  
    실행문 1;  
} else {  
    실행문 2;  
}
```

```
if(조건식1) {  
    실행문 1;  
} else if(조건식2) {  
    실행문 2;  
} else {  
    실행문 3;  
}
```

② 선택문 : switch 문은 변수에 저장된 값과 switch문에 있는 경우(case)의 값을 검사하여, 변수와 경우의 값에서 일치하는 값이 있을 때 그에 해당하는 실행문을 실행

```
var 변수=초기값;  
switch(변수) {  
    case 값1: 실행문1;  
    break;  
    case 값2: 실행문2;  
    break;  
    case 값3: 실행문3;  
    break;  
    case 값4: 실행문4;  
    break;  
    default: 실행문5;  
}
```

③ 반복문 : 일련의 실행문을 원하는 횟수만큼 반복하여 실행시킬 때 사용합니다.  
while 문과 do while 문, for 문이 있습니다.

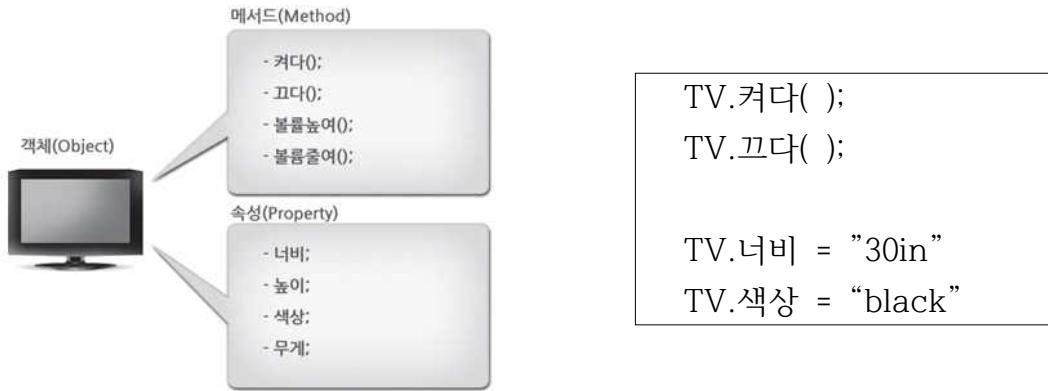
```
var 변수=초기값;  
while(조건식) {  
    실행문;  
    증감식;  
}
```

```
var 변수=초기값;  
do {  
    실행문;  
    증감식;  
} while(조건식)
```

```
for(초기값; 조건식; 증감식) {  
    실행문;  
}
```

#### 4. 객체

자바스크립트는 객체(Object) 기반 언어로, 객체는 기능 또는 속성으로 구성되어 있습니다.



① 자바스크립트 객체에 메서드와 속성 사용의 기본형입니다

```
객체(Object). 메서드();    //객체에 구성된 기능을 사용할 때.
객체(Object). 속성;    또는  객체(Object). 속성 = 값;
```

② 객체생성법 : 객체를 생성하는 기본 형식은 다음과 같이 new 키워드와 생성함수를 이용.  
이렇게 생성한 객체는 인스턴스네임(참조변수)으로 참조시켜 사용합니다.

③ 객체의 종류와 기능 : 자바스크립트의 객체는 **내장 객체**와 **브라우저 객체 모델**(BOM:Browser Object Model), **문서 객체 모델**(DOM:Document Object Model)로 나눌 수 있습니다.

(ㄱ) 내장객체 : 자바스크립트 자체 내에서 지원하는 객체를 가리킵니다. 내장 객체에는 문자 (String), 날짜(Date), 수학(Math), 배열(Array) 객체 등으로 구성되어 있습니다

날짜 객체

```
var 인스턴스네임(참조변수) = new Date( );    //현재 날짜 객체 생성
```

(ㄴ) 브라우저 객체 모델(BOM : Browser Object Model)

브라우저에 내장된 객체를 가리킵니다. 브라우저 객체의 최상위 객체는 window 객체이며 그 하위로는 screen, location, history, navigator 등으로 구성되어 있습니다.

window 객체

```
open( ),    alert( ),    prompt( ),    confirm( ),    moveTo( ),    resizeTo( )
setInterval(),    clearInterval(),    setTimeout( ),    clearTimeout( )
```

(ㄷ) 문서 객체 모델(DOM : Document Object Model)













문서 객체 모델이란 HTML 문서의 구조를 가리킵니다. <img> 태그에는 src라는 속성과 그림을 표현하는 기능을 가지고 있습니다. 그러므로 HTML 태그는 다른 말로 문서 객체라 부릅니다.

선택자

선택자는 자바스크립트를 이용해 HTML에 문서 객체를 선택할 때 사용합니다. 스타일(CSS)에 선택자와 비슷하다고 생각하면 되겠습니다.

```
CSS 선택자    =    javascript 선택자
```

(4-1) 자바스크립트 문자 객체

Method	Description
charAt(인덱스)	문자열에서 인덱스 번호에 해당하는 문자를 반환 var str="hello javascript"; str.charAt(1);  e
charCodeAt("찾을문자")	찾을 문자의 아스키코드 값을 반환한다.
concat("새로운문자")	문자열에 새로운 문자열을 결합 var str="hello"; str.concat("web");  hello web
fromCharCode(아스키코드값)	아스키 코드값에 해당하는 문자를 반환
indexOf("찾을문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 문자의 인덱스 번호를 반환. 만일 찾는 문자가 없으면 -1을 반환 var str="web he she he"; str.indexOf("he");  4
lastIndexOf("찾을문자")	문자열에서 오른쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 문자의 인덱스 번호를 반환. 만일 찾는 문자가 없으면 -1을 반환 var str="web he she he"; lastIndexOf("he");  11
length	문자열에서 문자의 개수를 반환 var str="hello javascript"; str.length;  16
match("찾을문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 찾은 문자를 반환. 만일 찾는 문자가 없으면 null을 반환 var str="web he she he"; str.match("boy");  null
replace("바꿀문자","새문자")	문자열에서 왼쪽부터 바꿀 문자와 일치하는 문자를 찾아 최초로 찾은 문자를 새 문자로 치환 var str="web he she he"; str.replace("web", "boy");  boy he she he
search("찾을문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 인덱스 번호를 반환 var str="web he she he"; str.search("he");  null hypothesis
slice(a, b)	a개의 문자를 자르고 b번째 이후에 문자를 자른 후 남은 문자를 반환 var str="hello javascript"; str.slice(3, 7);  lo j
split("문자")	지정한 문자를 기준으로 문자 데이터를 나누어 배열에 지정하여 변환 var str="webkmobilek200"; var srr=str.split("k"); 
substr(a, 문자개수)	문자열에 a 인덱스부터 지정한 문자 개수의 문자를 반환 var str="hello javascript"; str.substr(3, 4);  lo
substring(a, b)	a 인덱스부터 b 인덱스 이전 구간의 문자를 반환 var str="hello javascript"; str.substring(3, 7);  lo j
toLowerCase()	문자열에서 영문 대문자를 모두 소문자로 바꾼다.
toUpperCase()	문자열에서 영문 소문자를 모두 대문자로 바꾼다.
valueOf()	String 객체의 원시 값을 반환

<http://www.w3schools.com/js/default.asp>

## (4-2) 정규표현식이란

특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어. 문자열의 검색과 치환을 위해 지원

```
var 참조변수 = new RegExp(패턴, 검색옵션)
var 참조변수 = /패턴/검색옵션
```

정규표현 메서드

test() 지정된 규칙에 맞는 단어가 있으면 true 값 반환

exec() 지정된 규칙에 맞는 단어가 있으면 해당 단어 반환

### \* 정규 표현식

정규식	설명	예
\$	문자열이나 행의 끝에서 규칙에 맞는 문자 검색	
*	0번 이상 일치하는 문자를 검사할 때 사용	"a*b"는 a가 0번 이상 반복되므로 "b", "ab", "aab", "aaab"를 포함한다.
+	1번 이상 일치하는 문자를 검사할 때 사용	"a+b"는 a가 1번 이상 반복되므로 "ab", "aab", "aaab"를 포함하지만 "b"는 포함하지 않는다.
?	0번 또는 1번 규칙에 맞는 문자가 있는지 검사	"a?b"는 "b", "ab"를 포함하지만 "aab"는 포함하지 않는다.
i	찾는 문자가 영문일 때 대.소문자 구분하지 않음	
g	옵션에 g가 없으면 데이터의 왼쪽부터 규칙에 일치하는 문자 한 개만 찾지만, g를 넣으면 규칙에 맞는 모든 문자열을 찾는다.	
m	데이터의 행이 바뀌어도 규칙에 맞는 문자를 찾는다.	
^	문자열 앞에서부터 규칙에 일치하는 문자를 찾는다. 또는 부정을 나타냄	
.	임의의 한문자로 끝남	
{n}	n번 반복	
{n,}	n번 이상 반복	
{n,m}	최소n번, 최대m번 반복(n부터 m까지)	"a{1,3}b"는 "ab", "aab", "aaab"를 포함하지만, "b"나 "aaaab"는 포함하지 않는다.
[xy]	x, y중 하나	"[abc]d"는 ad, bd, cd를 포함. 또한, "-" 기호와 함께 쓰면 범위를 지정할 수 있다.
[^xy]	x,y를 제외한 문자 하나	"[^abc]d"는 ad, bd, cd는 포함하지 않고 ed, fd 등을 포함한다.
[x-z]	x-z사이의 문자 중 하나	"[a-c]d"는 ad, bd, cd를 포함. ed, fd 등은 포함하지 않는다.
x y	x 또는 y인지 검사할 때 사용	
(x)	[0~9] ()안의 내용을 그룹화	"abc adc"와 "a(b d)c"는 같은 의미이다.
\d	[^0~9] 0~9까지의 숫자 하나를 찾는다.	
\D	숫자를 제외한 문자 하나를 찾는다.	
\a	[A-Za-z] 알파벳 문자를 찾는다.	
\l	[a-z] 알파벳 소문자를 찾는다.	
\u	[A-Z]알파벳 대문자를 찾는다.	
\w	[a-zA-Z0-9_]알파벳+숫자+밑줄(_)를 찾는다.	
\W	[^a-zA-Z0-9_]알파벳+숫자+밑줄(_)가 아닌 문자를 찾는다.	
\x	[A-Fa-f0-9] 16진수	
\s	하나의 공백 문자(스페이스)를 찾는다.	
\S	공백문자가 아닌 문자를 찾는다.	
\n	줄바꿈	
\t	탭	

### (4-3) 자바스크립트에서 정규식 사용하기 예제

```

1 var num = 123;
2 var reg = /^[0-9]+$/g;
3 var rst = reg.test(num);
4 if(!rst){
5     alert("숫자만 입력해주세요.");
6     return false;
7 }

```

#### \* 자주 사용하는 정규식

분류	정규식
한글만	/^[ㄱ-ㅎ ㅏ-ㅣ 가-힣]\$/g
영어만	/^[a-zA-Z]\$/g
숫자만	/^[0-9]\$/g
한글 금지	/^[^ㄱ-ㅎ ㅏ-ㅣ 가-힣]\$/g
영문자 금지	/^[^a-zA-Z]\$/g
숫자 금지	/^[^0-9]\$/g
특수문자	/[W[W]{ }()<>? `~!@#\$\$%^&*~_+=,,:W"WWW'WW]/g
아이디 (영문소문자+숫자 4~12자리, 영문소문자로 시작)	/^[a-z][a-z0-9_]{3,11}\$/
비밀번호 (영문대소문자+숫자 6~18자리)	/^[A-Za-z0-9_]{6,18}\$/
이름	/^[가-힣]{2,4}\$/
전화번호('-' 포함)	/^Wd{2,3}-Wd{3,4}-Wd{4}\$/
이메일	/^[a-z][a-z0-9_]{3,11}@([a-zA-ZW.-]+)W.([a-zA-ZW.]{2,6})\$/
IP주소	/^Wd{1,3}W.Wd{1,3}W.Wd{1,3}W.Wd{1,3}\$/
대문자만	/^[A-Z]\$/g
소문자만	/^[a-z]\$/g
URL주소	/^(https?:W/W/)?([a-zA-ZW.-]+)W.([a-zA-ZW.]{2,6})([W/WwW.-]*)*W/?\$/

\* 정규식 테스트 해볼 수 있는 사이트 <http://www.regexr.com/>