

html5= html태그 + css + java script 기본

화면컨텐츠 + 시각적효과 + 동작

변수	number boolean string undefined-키워드, 변수타입, 변수값대체표현 object function ecmascript6 추가 Symbol
연산자	=== : 값 동일 타입 동일 == : 값 동일 10 / 3-> 3.3333 parseInt() ?:
조건문	if else switch case break
반복문	for do-while while
이동문	break continue
입력	prompt()->키보드문자열 .string confirm()->2개버튼 선택 boolean
출력	alert() console.log() document.write("<h1>....");

1 2 3 5 7

#### 4장 객체 함수 배열

##### 자바스크립트

ecmascript5- 객체 정의 생성자 호출

-객체= 연관 변수와 기능 집합

##### 자바스크립트 문법

1>객체를 literal 값 표현

회사원(id name salary)

```
var emp1 = {
```

```
  id : 100 ,
```

```
  name : "이사원" ,
```

```
  salary : 456000.99
```

```
};
```

```
emp1.dept = "인사부";//추가
```

```
if(id in emp1){// 변수 존재 여부 확인
```

```
delete emp1.id; //삭제
```

```
}
```

```
var emp2 = {
```

```
  id : 200 ,
```

```
name : "박대리" ,
salary : 556000.99
};
```

2>

3>

- 함수

객체=변수와 메소드 집합

함수: 1개 기능 집합

1>

```
function 함수명(매개변수){
    구현 내용;
//return 0;
}==> 함수 정의
```

==>함수 호출

2>

```
var f2=function(){};
```

- 객체의 메소드

var emp1 = { id:100 , , ... 함수.}	메소드 - 객체 내부 포함 함수
함수(){ ---> Window내장객체 (모든 영역 모두 사용 가능. alert() )	전역객체 포함 함수 함수  메소드 + 함수 정의 문법 동일

<b>var a = function(){</b> 무명의 함수 리터럴 표현 함수 <b>a();</b>	<b>function b(){}</b> 유명의 함수	내장함수=전역함수 <b>parseInt/parseFloat/</b> <b>alert/prompt/confirm</b>
<b>var a = {}</b> 리터럴 표현 객체 <b>a.xxxxx()</b>	<b>function Product(code, name){</b> <b>  this.code= code;</b> <b>  this.name = name;</b> <b>  this.xxx                       =</b> <b>  function(){ this.code }</b> <b>  }</b> <b>var p1 = new Product(1, "컴퓨터")</b> <b>p1.code==&gt;ok</b> <b>p1.name==&gt;ok</b>	내장객체 <b>Number</b> <b>String</b> <b>Array</b> <b>Date</b> <b>Function-&gt; 함수정의3번째방법</b> <b>Canvas</b>

- 자바스크립트 객체

1. 사용자 정의 객체 1-1.리터럴 1-2.생성자 정의 호출		
2. 내장객체 ecmascript 표준 객체	3. 내장 표준 아니다 웹브라우저 정보 표현 객체 <b>Window</b> <b>parseInt / parseFloat</b> <b>alert/confirm/prompt</b> <b>xxxxxx</b> <b>Navigator</b> <b>Screen</b> <b>History</b> <b>DOM</b>	

## -클래스 정의-18장

### 8장

#### - 함수 정의 + 호출

1>function a(){ } - a();

2>var b = function(){ } - b()

var c = {};

c.f = function(a){...} - c.f(0);

#### -callback 함수

##### - 다른 함수에 매개변수로 전달하는 함수

- setTimeout( 함수명 , 시간);

- setTimeout( function(){ } , 시간);

- 함수를 변수로 취급

- 매개변수로 함수를 사용 가능

3>var c = new Function("매개변수", "구현내용");

4>ecmascript 6 – 화살표 함수-react

자바스크립트 화살표함수 (int x)=>{.....}(10);

## 6장 입출력 / parseInt / 이벤트 처리 타이머 캔버스