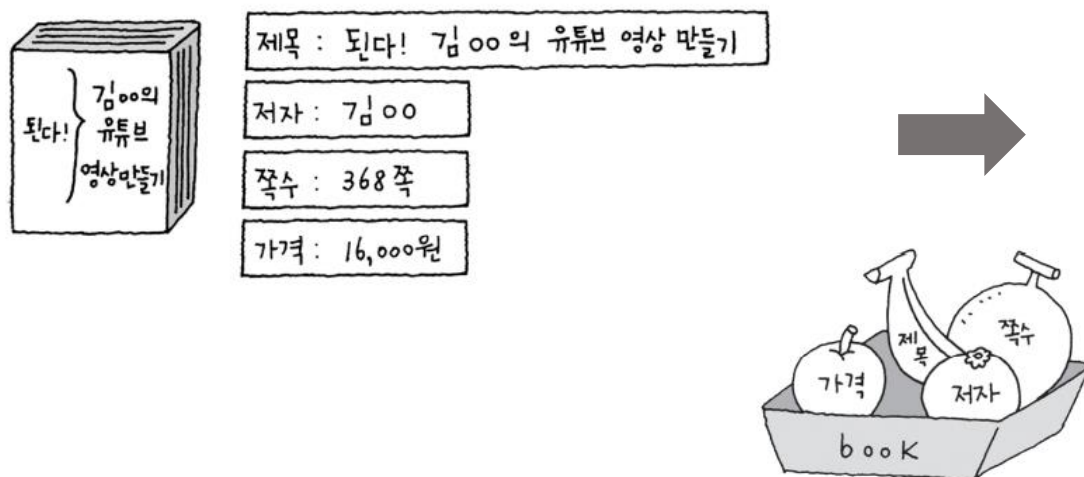


06. 객체

객체란

- 객체는 자료를 저장하고 처리하는 기본 단위
- 자바스크립트 프로그램에서 인식할 수 있는 모든 대상
- 복합 자료형 (객체 안에 숫자, 문자열 등 여러 가지 자료형이 포함됨)



```
> var book = {  
    title: "자바스크립트", // 제목  
    author: "홍길동",      // 저자  
    pages: 500,            // 쪽수  
    price: 15000           // 가격  
}
```

객체란

내장 객체(built-in object) - Number, Boolean, Array, Math 등

- 미리 정의되어 있는 개체

- 1) 문서 객체 모델(DOM) : 문서 뿐만 아니라 웹 문서 안에 포함된 이미지·링크·텍스트 필드 등을 모두 별도의 객체로 관리 (이미지, 텍스트, 링크 등)
- 2) 브라우저 객체 모델 : 웹 브라우저 정보를 객체로 관리 (Navigator, History 등)

사용자 정의 객체

- 필요할 때마다 사용자가 직접 만드는 객체 (book)

객체란

속성(프로퍼티, property)

- 객체에 담고 있는 값
- 객체명.속성명

(예, Navigator 객체에 있는 vendor 속성이 궁금하다면? navigator.vendor)

메서드(method)

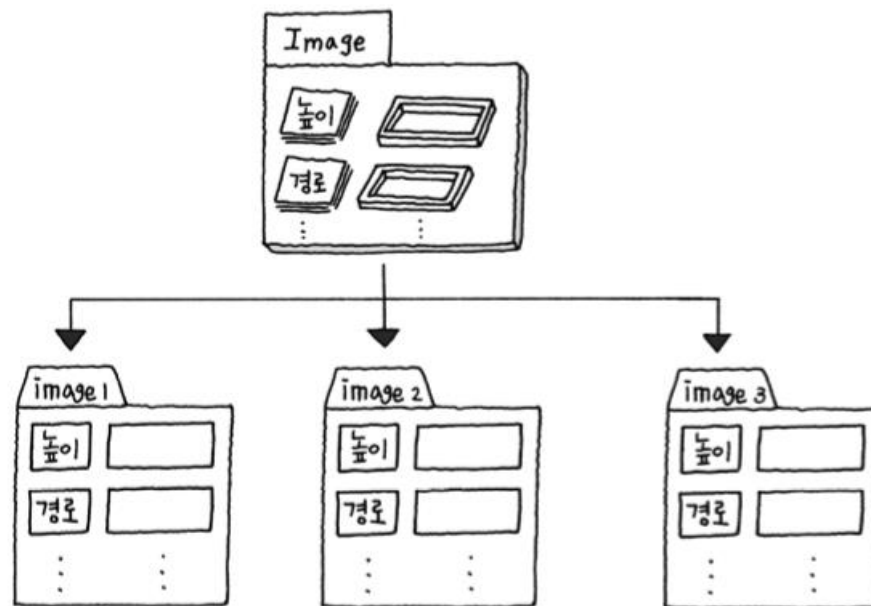
- 객체의 동작을 정의한 함수 (객체 안의 함수, f 로 줄여서 사용)
- Window 객체에는 알림 창을 표시하는 alert() 메서드가 있다
 - window.alert()
 - window 객체는 최상위 객체이므로 window. 생략하고 그냥 alert()라고 쓸 수 있음.

```
> window
< Window {window: Window, self: Window, document: document, name: '', location: Location, ...}
  ▼ alert: f alert()
    length: 0
    name: "alert"
    arguments: (...)
    caller: (...)
    ▶ [[Prototype]]: f ()
```

객체란

프로토타입과 인스턴스

- 프로토타입 : 객체의 속성과 메서드를 모아놓은 틀 같은 것
ex) 웹문서에 3개의 이미지를 포함하여야 한다면, Image 객체를 이용해서 똑같은 3개를 사용할 수 있음
- 인스턴스 : 프로토타입으로 찍어낸 객체.
프로토타입의 속성과 함수를 똑같이 사용함.
- ex) 프로토타입을 통해 생성된 객체, image1 image2 image 3



객체 인스턴스 만들기

- new 예약어 뒤에 괄호를 붙여서 인스턴스 만듦.
- 예) 날짜나 시간에 관련된 프로그램을 만들려면 날짜/시간 정보가 있는 Date 객체의 인스턴스부터 만들어야 함.

② now 변수에 저장

① Date 객체의 인스턴스를 만들고

```
> var now = new Date( )
> now
< Fri Nov 02 2018 21:50:52 GMT+0900 (한국 표준시)
```

- new 예약어로 Date 객체 인스턴스 생성
- now 변수에 저장한 후 now 변수 값을 콘솔창에 출력

```
> now.toLocaleString( )
< "2018. 11. 2. 오후 9:51:33"
```

- now는 이제 Date 객체의 인스턴스이기 때문에 Data 객체에서 정의한 속성과 함수를 모두 사용할 수 있음
- Data객체의 toLocaleString() 함수 사용
 - toLocaleString() : 로컬(현재 지역)에 맞게 날짜/시간 변환

Math 객체를 사용해 무작위 수 만들기

random() 메서드

0과 1 사이의 무작위 수

```
> Math.random( )  
< 0.8693164091808827  
> Math.random( )  
< 0.9967725951733857  
> Math.random( )  
< 0.38758779834374035  
> Math.random( )  
< 0.3176968955282209
```



1과 100 사이의 수를 뽑는다면?

$0.0000 \sim 99.9999 + 1$

```
> Math.random( ) * 100 + 1  
< 69.77507422327689
```



정수가 필요하다면?

floor() 메서드 - 소수점 이하 버림

```
> Math.floor(Math.random( ) * 100 + 1)  
< 40  
> Math.floor(Math.random( ) * 100 + 1)  
< 57  
> Math.floor(Math.random( ) * 100 + 1)  
< 38  
> Math.floor(Math.random( ) * 100 + 1)  
< 70
```

사용자 정의 객체

리터럴 표기법을 사용한 객체 만들기

```
> var book = {  
  title: "자바스크립트",  
  author: "고쌤",  
  pages: 500,  
  price: 15000,  
  info: function( ) {  
    alert(this.title + "책의 분량은 " + this.pages + "쪽입니다.");  
  }  
};
```

객체

속성 이름

속성 값

함수 이름

함수

book 객체의
속성과 함수 사용하기

```
> book.title  
< "자바스크립트"  
> book.info( )
```

book 객체에 새로운
속성 추가하기

```
> book.field = "IT"  
< "IT"
```


생성자 함수를 사용해 객체 만들기

```
> function Book(author, pages, price, title) {  
  this.author = author;  
  this.pages = pages;  
  this.price = price;  
  this.title = title;  
}
```

생성자 함수의 this는 객체 자신(Book)을 가리킴

도서관에서 책을 관리하는 객체가 필요하다면

생성자 함수를 통해 속성과 틀은 미리 만들어두고,
필요에 따라 틀을 복제하여 인스턴스로 생성

Book 객체의 인스턴스 만들기

```
> jsBook = new Book("홍길동", 500, 15000, "자바스크립트") //인스턴스 만들기  
< ▶ Book {author: "홍길동", pages: 500, price: 15000, title: "자바스크립트"}  
> jsBook.title //jsBook 객체에서 속성 값 확인하기  
< "자바스크립트"
```

[미리보기] 기념일 계산기 만들기

우리 만난 지	
34일	
기념일 계산	
100일	2018년 6월 1일
200일	2018년 9월 9일
1년	2019년 2월 23일
500일	2019년 7월 5일

- 특정일로부터 오늘까지 며칠이 지났는지
- 특정일로부터 100일, 1년 되는 날이 언제인지
계산하는 프로그램

날짜와 시간 정보를 가지는 Date 객체

여러 방법으로 인스턴스 만들기

> new Date()

> new Date("2018-02-25")

> new Date("2018-02")

> new Date("2018-02-25T18:00:00")

<참고> 자바스크립트의 날짜/시간 입력 방식

1) ISO 형식 (YYYY-MM-DDTHH:MM:SS)

2018-02-25T18:00:00

2) 짧은 날짜 형식(MM/DD/YYYY)

02/25/2018

3) 긴 날짜 형식(MM DD YYYY)

Thu Aug 17 2017 15:00:41 GMT+0900

Date 객체의 메서드

날짜/시간 정보를 가져오는 함수

함수 이름	설명
getFullYear()	날짜 정보에서 연도(Year) 정보를 가져와 네 자리 숫자로 표시합니다.
getMonth()	날짜 정보에서 '월(Month)' 정보를 가져옵니다. 이때 0~11의 숫자로 월을 표시합니다. 0부터 1월이 시작되고 11은 12월입니다. 예 0: 1월, 1: 2월, ..., 10: 11월, 11: 12월
getDate()	날짜 정보에서 며칠인지 알 수 있는 '일(Date)' 정보를 가져옵니다.
getDay()	날짜 정보에서 '요일(Day)' 정보를 가져옵니다. 이때 요일 정보는 0~6의 숫자로 표시되는데, 0은 '일요일'에 해당하고 6은 '토요일'에 해당합니다. 예 0: 일요일, 1: 월요일, 2: 화요일, 3: 수요일, 4: 목요일, 5: 금요일, 6: 토요일
getTime()	1970년 1월 1일 자정 이후의 시간을 밀리초로 표시합니다. 밀리초는 1/1000 초를 가리킵니다.
getHours()	0~23의 숫자로 시를 표시합니다.
getMinutes()	0~59의 숫자로 분을 표시합니다.
getSeconds()	0~59의 숫자로 초를 표시합니다.
getMilliseconds()	0~999의 숫자로 밀리초를 표시합니다.

Date 객체의 메서드

날짜/시간 정보를 설정하는 함수

함수 이름	설명
setFullYear()	연도를 네 자리 숫자로 설정합니다.
setMonth()	0~11의 숫자로 월을 표시합니다. 0부터 1월이 시작되고 11은 12월입니다.
setDate()	1~31의 숫자로 일을 설정합니다.
setTime()	1970년 1월 1일 자정 이후의 시간을 밀리초로 설정합니다.
setHours()	0~23의 숫자로 시를 설정합니다.
setMinutes()	0~59의 숫자로 분을 설정합니다.
setSeconds()	0~59의 숫자로 초를 설정합니다.
setMilliseconds()	0~999의 숫자로 밀리초를 설정합니다.

Date 객체의 메서드

콘솔창에서 실행해보세요.

```
var now = new Date() // Date 객체의 인스턴스  
now  
Mon Nov 05 2018 13:58:45 GMT+0900 (한국 표준시)
```

```
now.getDate() // 날짜 정보만  
5
```

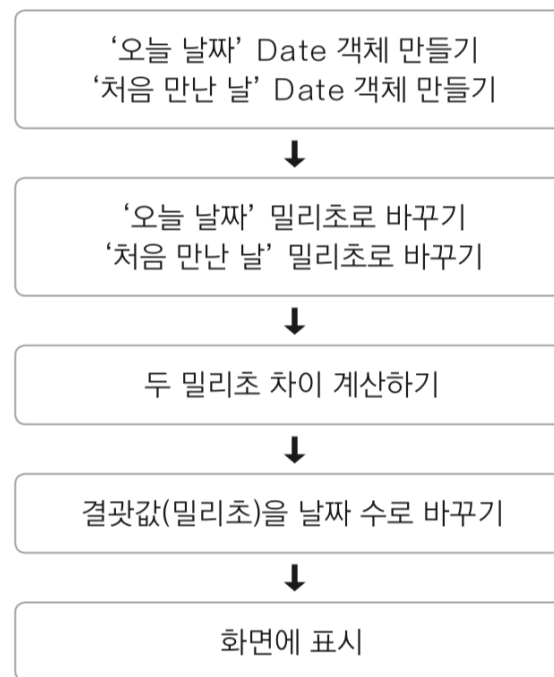
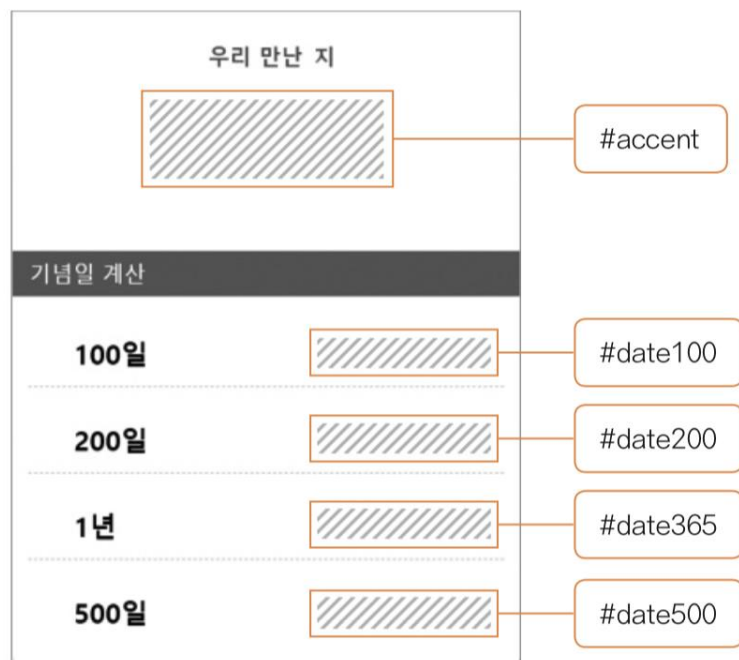
```
now.getDay() // 요일 정보만  
1
```

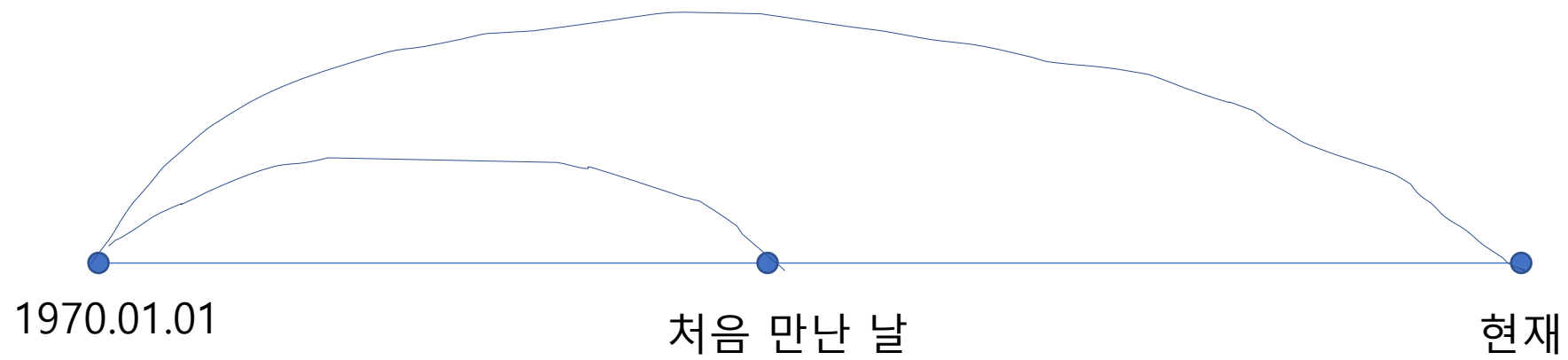
```
now.getTime() // 1910년 1월 1일 00:00을 기준으로 밀리초  
1545713925796
```

```
now.setDate(now.getDate() + 50) // 현재 날짜에서 + 50일  
now  
Tue Dec 25 2018 13:58:45 GMT+0900 (한국 표준시)
```

기념일 계산기 만들기

실습 파일 : 06Wdday.html





며칠 지났는지 알아보기

실습 파일 : 06Wdday.html

```
1  var now = new Date(); // 오늘 날짜 정보를 Date 객체의 인스턴스 now 객체로 만듭니다.
2  var firstDay = new Date("2018-03-23"); // 처음 만난 날의 날짜 정보를 firstDay 객체로 만듭니다.
3  var toNow = now.getTime(); // 오늘 날짜를 밀리초로 바꿉니다.
4  var toFirst = firstDay.getTime(); // 처음 만난 날을 밀리초로 바꿉니다.
5  var passedTime = toNow - toFirst; // 처음 만난 날과 오늘 사이의 차이 (밀리초)
6  var passedDay = Math.round(passedTime/(1000*60*60*24)); // 밀리초를 일로 변환 후 반올림합니다.
7  document.querySelector('#accent').innerText = passedDay + "일"; // #accent 영역에 표시합니다.
```

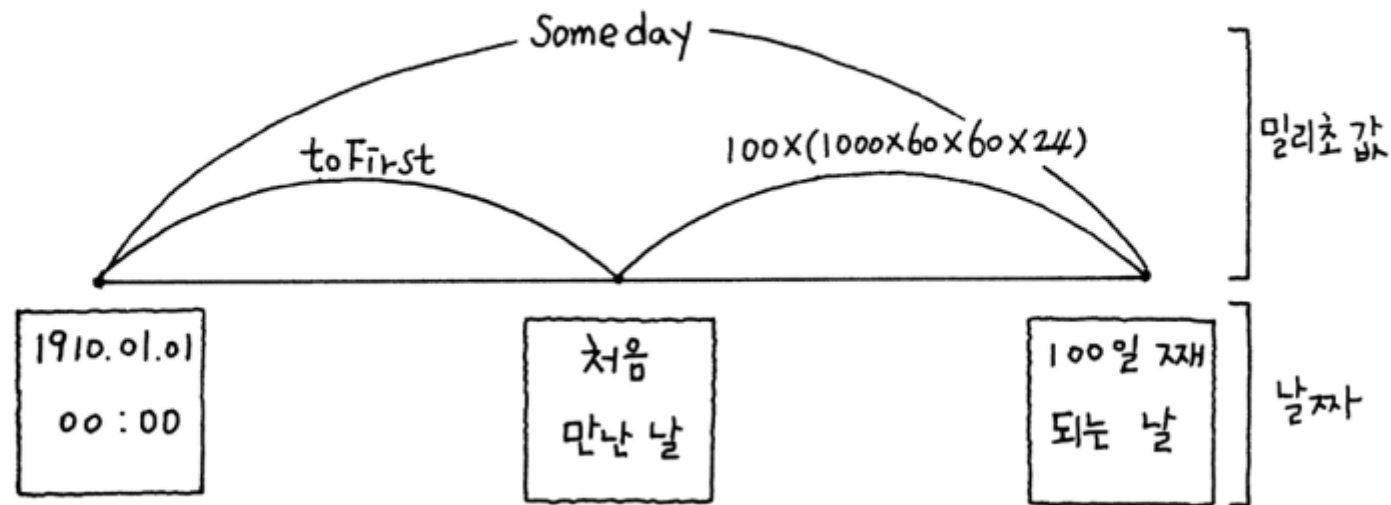
100일 후 날짜, 200일 후 날짜 계산하기

실습 파일 : 06Wdday.html

```
9  var future = toFirst + 100*(1000*60*60*24); // 처음 만난 날에 밀리초로 바꾼 100일을 더합니다.
10 var someday = new Date(future); // future100 값을 사용해 Date 객체의 인스턴스를 만듭니다.
11 var year = someday.getFullYear( ); // '연도'를 가져와 year 변수에 저장합니다.
12 var month = someday.getMonth( ); // '월'을 가져와 month 변수에 저장합니다.
13 var date = someday.getDate( ); // '일'을 가져와 date 변수에 저장합니다.
14 document.querySelector("#date100").innerText = year + "년 " + month + "월 " + date + "일";
15
16 future = toFirst + 200*(1000*60*60*24); // 처음 만난 날에 밀리초로 바꾼 200일을 더합니다.
17 someday = new Date(future); // future 값을 사용해 Date 객체의 인스턴스를 만듭니다.
18 year = someday.getFullYear( ); // '연도'를 가져와 year 변수에 저장합니다.
19 month = someday.getMonth( ); // '월'을 가져와 month 변수에 저장합니다.
20 date = someday.getDate( ); // '일'을 가져와 date 변수에 저장합니다.
21 document.querySelector("#date200").innerText = year + "년 " + month + "월 " + date + "일";
```

100일 후 날짜, 200일 후 날짜 계산하기

실습 파일 : 06Wdday.html



calcDate() 함수 선언하기

실습 파일 : 06Wdday.html

```
9  calcDate(100); // 100일 기념일을 계산해서 표시합니다.
10 calcDate(200); // 200일 기념일을 계산해서 표시합니다.
11 calcDate(365); // 1년 기념일을 계산해서 표시합니다.
12 calcDate(500); // 500일 기념일을 계산해서 표시합니다.
13
14 function calcDate(days) {
15     var future = toFirst + days*(1000*60*60*24); // 처음 만난 날에 밀리초로 바꾼 100일을 더합니다.
16     var someday = new Date(future); // future 값을 사용해 Date 객체의 인스턴스를 만듭니다.
17     var year = someday.getFullYear( ); // '연도'를 가져와 year 변수에 저장합니다.
18     var month = someday.getMonth( ); // '월'을 가져와 month 변수에 저장합니다.
19     var date = someday.getDate( ); // '일'을 가져와 date 변수에 저장합니다.
20     document.querySelector("#date"+days).innerText = year + "년 " + month + "월 " + date + "일";
21 }
```