훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체 모델링

◆ 객체

- 세상에 존재하는 모든 것

◆ 자동차 객체의 모델링

객체	속성	메소드
car	car.name="Sonata" car.speed=100 car.color="white" car.door=4	car.start( ) { } car.accel( ) { } car.break( ) { } car.transe( ) { }

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 객체

◆ 자바스크립트 객체

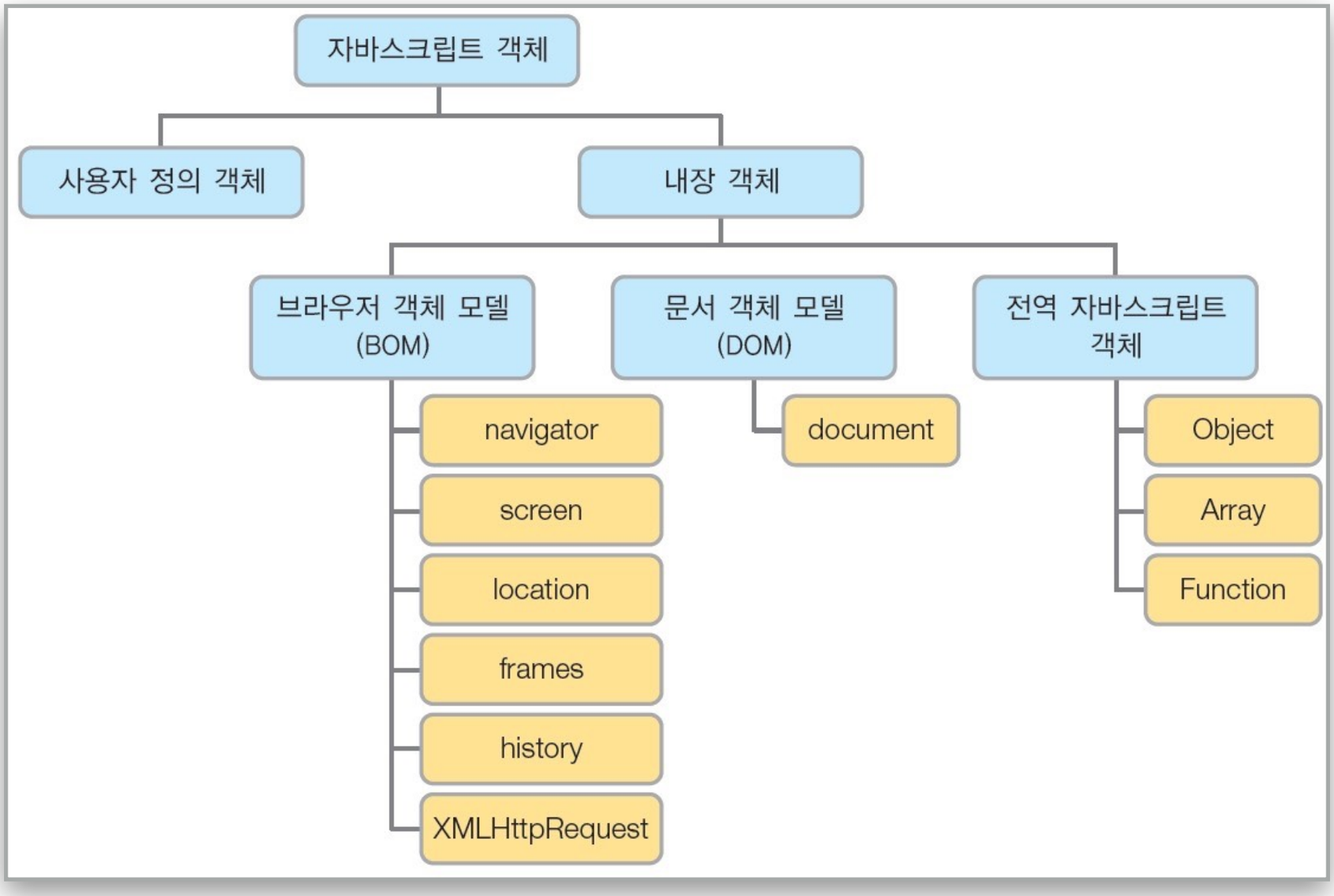
- 사용자 정의 객체: 사용자가 직접 객체의 속성과 메소드를 정의하여 사용하는 객체
  (예: Car( ), House( ), Hotel( ))
- 내장 객체: 자바스크립트 프로그램 자체에서 정의하여 사용자에게 제공하는 객체
  (예: Object( ), Array( ), Date( ))

◆ 내장 객체의 종류

- 브라우저 객체 모델(BOM, Browser Object Model): 웹 브라우저의 각종 요소를 객체로 표현
- 문서 객체 모델(DOM, Document Object Model): 웹 문서의 각종 요소를 객체로 표현
- 전역 자바스크립트 객체(Global JavaScript Objects): 자바스크립트 프로그램 전체에서 사용하는 내장 객체

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 객체

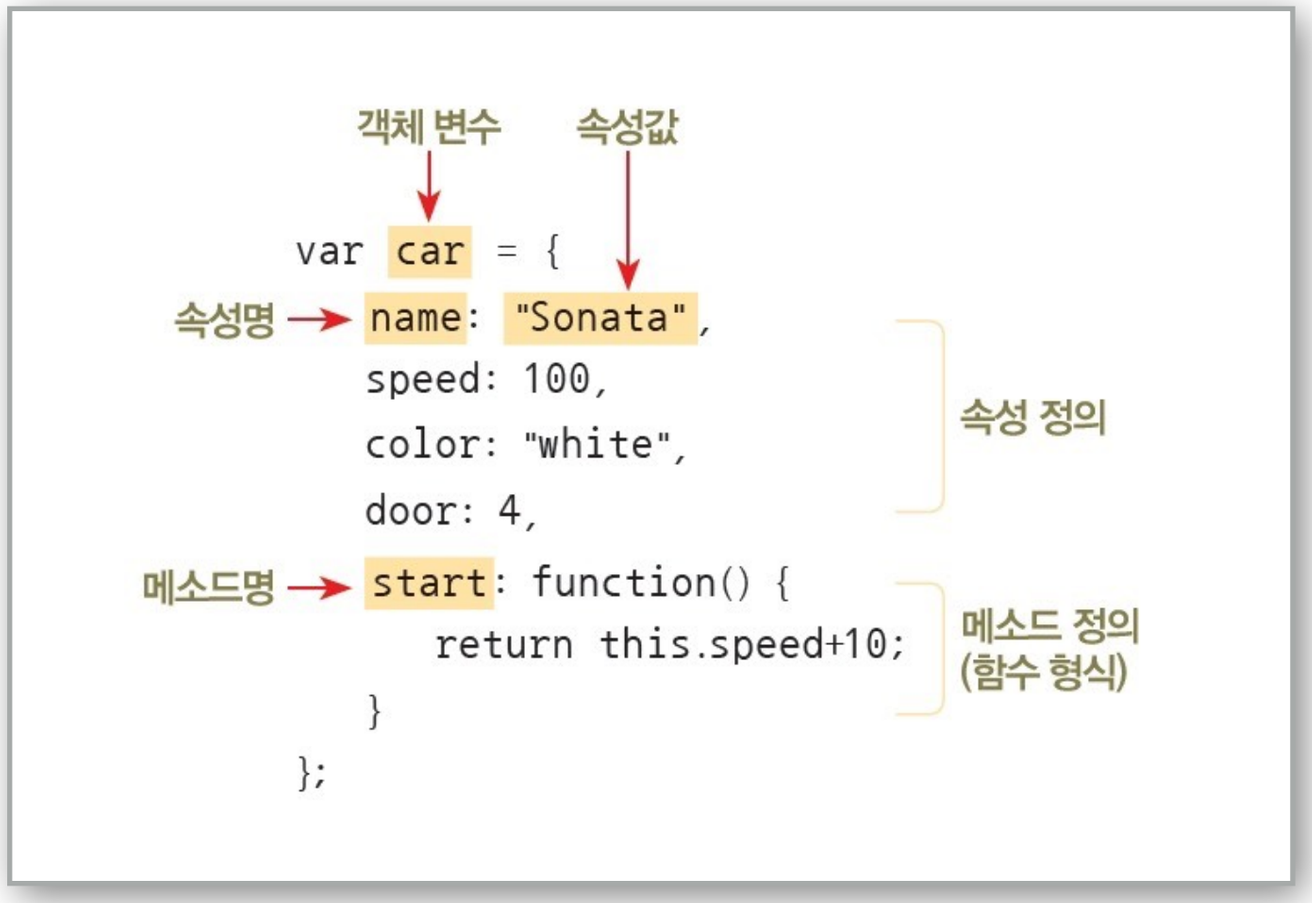


◆ 자바스크립트 객체의 종류

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체 변수 이용 방법

◆ 객체 변수를 이용하여 객체 생성



◆ 객체 속성 접근 방법

방법	사용 예	방법	사용 예
객체명.속성명	car.name car.speed car.color	객체명['속성명']	car['name'] car['speed'] car['color']

❖ 객체 변수 이용 방법

◆ 자바스크립트로 제어할 요소를 찾아 결과를 출력하는 방법

방법	사용 예	의미
innerHTML 속성 이용	document.getElementById("carname").innerHTML;	웹 문서 안에서 아이디가 "carname"인 요소를 찾아 내용을 출력한다.
textContent 속성 이용	var cname=document.getElementById("carname"); cname.textContent;	웹 문서 안에서 아이디가 "carname"인 요소를 찾아서 cname 변수에 반환한 후 cname 변수의 내용을 출력한다.

◆ 속성만 가진 객체 만들기

js_obj01.html

```
<body>
   <p id="var1"></p>
   <p id="var2"></p>
   <p id="var3"></p>
   <script>
      var car={name: 'Sonata', speed: 100, color: 'white'};
      document.getElementById("var1").innerHTML="자동차 이름 : " + car['name'];
      document.getElementById("var2").innerHTML="자동차 속도 : " + car.speed;
      document.getElementById("var3").innerHTML="자동차 색상 : " + car.color;
   </script>
</body>
```

❖ 객체 변수 이용 방법

◆ 메소드를 호출하여 연산 결과 출력하기

js_obj02.html

```
<body>
    <p id="msg1"></p>
    <p id="msg2"></p>
    <p id="msg3"></p>
    <script>
        var obj={
            m1: function() {
                return "Hello Sonata";
            },
            m2: function(a) {
                var result=a;
                return result;
            },
            m3: function(a, b) {
                var result=a+b;
                return result;
            }
        };
        document.getElementById("msg1").innerHTML=obj.m1();
        document.getElementById("msg2").innerHTML=obj.m2(100);
        document.getElementById("msg3").innerHTML=obj.m3(100, 200);
    </script>
</body>
```

★ 객체 맴버함수 만들기 2

```
함수이름 : function() {
        return data;
}
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체 변수 이용 방법

◆ 자동차 객체 생성하기

js_obj03.html

```
<body>
    <p id="carname"></p>
    <p id="carcolor"></p>
    <p id="carspeed"></p>
    <script>
        var car={
            name: 'Sonata',
            speed: 50,
            color: 'white',
            start: function() {
                return this.speed+10;
            }
        };
        var cname=document.getElementById("carname");
        cname.textContent=car.name;
        var colname=document.getElementById("carcolor");
        colname.textContent=car.color;
        var cspeed=document.getElementById("carspeed");
        cspeed.textContent=car.start();
    </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체 변수 이용 방법

◆ 자동차의 속도 조절하기

js_obj04.html

```
<body>
    <p id="upspeed"></p>
    <p id="downspeed"></p>
    <script>
        var car={
            name: 'Sonata',
            speed: 50,
            color: 'white',
            speedup: function() {
                return this.speed+10;
            },
            speeddown: function() {
                var low=this.speed-10;
                return low;
            }
        };
        var upspeed=document.getElementById("upspeed");
        upspeed.textContent='속도 증가 : ' + car.speedup();
        var downspeed=document.getElementById("downspeed");
        downspeed.textContent='속도 감소 : ' + car.speeddown();
    </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체 변수 이용 방법

◆ 자동차의 속도 제어하기

js_obj05.html

```
<body>
    <p id="upspeed"></p>
    <p id="downspeed"></p>
    <script>
        var car={
            name: 'Sonata',
            speed: 100,
            color: 'white',
            speedup: function(a) {
                var sp=this.speed+a;
                if(sp>=300) {
                    sp=50;
                    return sp;
                }
                else {
                    return sp;
                }
            },
            speeddown: function(a) {
                var sp=this.speed-a;
                if(sp<0) {
                    sp=0;
                    return sp;
                }
                else {
                    return sp;
                }
            }
        };
        var upspeed=document.getElementById("upspeed");
        upspeed.textContent='속도 증가 : ' + car.speedup(100);
        var downspeed=document.getElementById("downspeed");
        downspeed.textContent='속도 감소 : ' + car.speeddown(30);
    </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 생성자 함수를 이용하는 방법

◆ Object 함수 이용

```
<script>
        var car = new Object();                         // 객체 생성
        car.name = 'Lamborghini';                       // 속성 정의
        car.speed = 250;
        car.color = 'orange';
        car.speedup = function() {                      // 함수 정의
                return this.speed + 50 ;
        };
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 생성자 함수를 이용하는 방법

◆ Object 함수 이용 객체 만들기

js_obj06.html

```
<body>
    <p id="upspeed"></p>
    <p id="downspeed"></p>
    <script>
        var car={
            name: 'Sonata',
            speed: 50,
            color: 'white',
            speedup: function() {
                return this.speed+10;
            },
            speeddown: function() {
                var low=this.speed-10;
                return low;
            }
        };
        var upspeed=document.getElementById("upspeed");
        upspeed.textContent='속도 증가 : ' + car.speedup();
        var downspeed=document.getElementById("downspeed");
        downspeed.textContent='속도 감소 : ' + car.speeddown();
    </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 생성자 함수를 이용하는 방법

◆ 생성자 함수 정의

```
function Car(name, color, speed) {
        this.name = name;
        this.color = color;
        this.speed = speed;
        this.speedup = function(){
                return this.speed + 10;
        };
        this.speeddown = function() {
                return this.speed - 10;
        }
}
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 생성자 함수를 이용하는 방법

- ◆ 생성자 함수 정의 후 객체 만들기

js_obj07.html

```
<body>
   <p>[Hong's Car]</p>
   <p id="carname"></p>
   <p id="carcolor"></p>
   <p id="carspeed"></p>
   <p>[Kim's Car]</p>
   <p id="carname2"></p>
   <p id="carcolor2"></p>
   <p id="carspeed2"></p>
   <script>
      function Car(name, color, speed) {
         this.name=name;
         this.color=color;
         this.speed=speed;
         this.speedup=function() {
            return this.speed+10;
         };
         this.speeddown=function() {
            return this.speed-10;
         };
      }
      var Hongcar=new Car('Sonata', 'blue', 100);
      var Kimcar=new Car('Jeep', 'red', 70);
      var cname=document.getElementById("carname");
      cname.textContent='자동차 이름 : ' + Hongcar.name;
      var colname=document.getElementById("carcolor");
      colname.textContent='자동차 색상 : ' + Hongcar.color;
      var cspeed=document.getElementById("carspeed");
      cspeed.textContent='자동차 속도 : ' + Hongcar.speedup();
      var cname=document.getElementById("carname2");
      cname.textContent='자동차 이름 : ' + Kimcar.name;
      var colname=document.getElementById("carcolor2");
      colname.textContent='자동차 색상 : ' + Kimcar.color;
      var cspeed=document.getElementById("carspeed2");
      cspeed.textContent='자동차 속도 : ' + Kimcar.speedup();
   </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 생성자 함수를 이용하는 방법

- ◆ 이미 생성된 객체에
  속성 추가및 삭제하기

js_obj08.html

```
<body>
   <p>[Hong's Car]</p>
   <p id="hong1"></p>
   <p id="hong2"></p>
   <p id="hong3"></p>
   <p>[Kim's Car]</p>
   <p id="data1"></p>
   <p id="data2"></p>
   <p id="data3"></p>
   <p id="data4"></p>
   <script>
      function Car(name, color, speed) {
         this.name=name;
         this.color=color;
         this.speed=speed;
         this.speedup=function() {
            return this.speed+10;
         };
         this.speeddown=function() {
            return this.speed-10;
         };
      }
      var Hongcar=new Car('Sonata', 'blue', 100);
      var Kimcar=new Car('Jeep', 'red', 70);
      Kimcar.price='3천만 원';
      delete Kimcar.color;
      var cname=document.getElementById("hong1");
      cname.textContent='자동차 이름 : ' + Hongcar.name;
      var colname=document.getElementById("hong2");
      colname.textContent='자동차 색상 : ' + Hongcar.color;
      var cspeed=document.getElementById("hong3");
      cspeed.textContent='자동차 속도 : ' + Hongcar.speedup();
      var cname=document.getElementById("data1");
      cname.textContent='자동차 이름 : ' + Kimcar.name;
      var colname=document.getElementById("data2");
      colname.textContent='자동차 색상 : ' + Kimcar.color;
      var cspeed = document.getElementById("data3");
      cspeed.textContent='자동차 속도 : ' + Kimcar.speedup();
      var cspeed=document.getElementById("data4");
      cspeed.textContent='자동차 가격 : ' + Kimcar.price;
   </script>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 객체에 함수 추가하기

```
객체이름.함수이름 = function([매개변수]) {
        . . . 함수 실행 문 . . .
        [return data값;]
};
```

❖ 함수 안에서 객체의 자원 활용하기

```
객체이름.함수이름 = function([매개변수]) {
        this.변수이름 = data값;
        var 변수이름 = this.함수이름(매개변수);
        [return data값;]
};
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 내장 객체

◆ 자바스크립트 내장객체란?

- 모든 웹 사이트에는 공통적으로 필요한 기능들이 존재한다.
  이 기능들을 매번 새로 만들어야 한다면 매우 비효율적이다.
- 이런 불편함 해소를 위해 자바스크립트에서 미리 마련해 둔 내장된 기능을 제공
  (예] 내장함수들…)

공통 기능의 필요성

- 웹페이지를 제작하는데 필요한 기능들 중에서
  대다수의 기능은
  모든 사이트가 공통적으로 요구하는 내용
- 모든 웹 퍼블리셔들이
  공통적인 기능을 개별적으로 제작한다면
  개발 시간에 대한 낭비를 초래

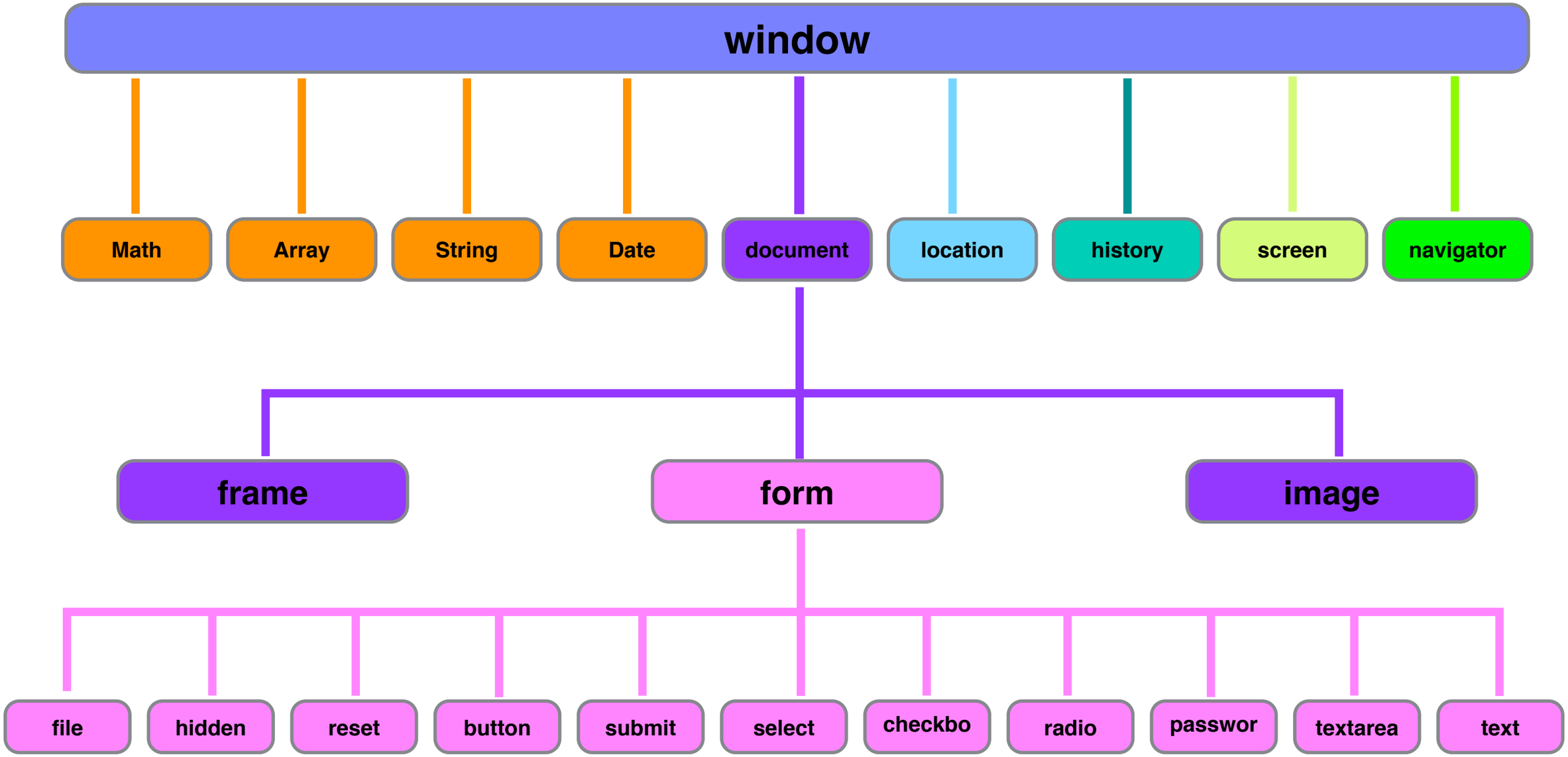
내장객체의 제공

- 모든 웹 브라우저 개발사들이
  이러한 공통 기능의 필요성을 인지하고,
  사전에 객체이름과
  함수 이름들을 통일하여 구현한뒤,
  브라우저 안에 내장
- 웹 퍼블리셔들은
  이러한 객체를 활용하는 것 만으로
  많은 기능들을 직접 구현하는 수고를 덜 수 있다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 내장 객체

◆ Javascript에서 제공되는 내장객체는 모두 window라는 내장객체의 하위 객체 형태로 존재



✻ 내장객체의 종류

❖ 자바스크립트 내장 객체

◆ 값의 처리를 위한 내장 객체

이 름	설 명
Date	시스템의 현재 날짜, 시각을 조회하거나 계산하기 위한 기능 제공
Array	같은 종류의 변수를 하나로 묶기 위한 배열에 관련된 기능 제공
String	하나의 문자열을 독립된 객체로 생성 문자열 안에서 특정 글자가 시작하는 위치, 문자열 안에서 원하는 내용만 추출하는 등의 기능 제공
Math	삼삭함수, 지수, 로그 등 수학과 관련된 각종 고급 함수를 제공 (일반적으로 잘 사용되지 않는다.)

❖ 자바스크립트 내장 객체

◆ 브라우저의 제어를 위한 내장 객체

이 름	설 명
window	브라우저 창에 대한 모든 상황을 제어하는 최상위 객체 모든 브라우저 제어관련 내장 객체는 Window객체의 하위에 존재
location	URL 정보를 제어하는 객체 페이지 이동, 현재 주소 조회, 새로 고침 등의 기능 제공
history	웹 브라우저에 기록되어 있는 히스토리 정보를 제어
navigator	브라우저의 종류를 판별
screen	브라우저 화면에 대한 정보를 알려준다. 변수값만 포함하고 있으며, 함수는 포함하고 있지 않다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 내장 객체

◆ HTML문서를 제어하기 위한 내장 객체

이 름	설 명
document	문서에 대한 정보, 즉 HTML 문서의 각 요소들을 제어하기 위한 기능
image	<img> 태그에 대한 속성을 제어하는 객체
form	입력양식 컴포넌트를 위한 개별 객체들을 포함
frame	웹 페이지 안에 다른 웹 페이지를 포함하는 Frameset과 iframe을 제어하는 기능 제공

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 내장 객체

◆ String 객체

* 문자 객체(String Object)는 문자형 데이터를 객체로 취급하는것

* 자바스크립트에서 가장 많이 사용

* String 객체 맴버 변수

이 름	설 명
length	문자열의 길이를 조회

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트 내장 객체

◆ String 객체

* String 객체 맴버 함수

이 름	설 명
String(value)	문자열을 생성하기 위한 생성자
String charAt(int)	지정된 위치의 글자를 리턴
int indexOf(String)	문자열 앞에서부터 파라미터로 주어진 글자를 검색하여 위치를 알려준다. 검색결과가 없을 경우 -1을 리턴
int lastIndexOf(String)	문자열 뒤에서 부터 파라미터로 주어진 글자를 검색하여 위치를 알려준다. 검색된 글자의 위치는 앞에서 부터 카운트 검색 결과가 없을 경우 -1 리턴
String substring(int, int)	문자열에서 첫 번째 파라미터의 위치부터 두 번째 파라미터의 위치까지 추출 두 번째 파라미터가 없을 경우 끝까지 추출
String toUpperCase()	대문자로 변환
String toLowerCase()	소문자로 변환
String slice(int, int)	첫 번째 파라미터 만큼 문자를 자르고 두번째 파라미터 이후의 문자를 자른후 남은 문자 반환
String replace(String, String)	첫번째 파라미터의 문자를 찾아 두번째 파라미터로 바꾼 후 반환
String match(String)	파라미터의 문자를 찾아 최초 찾은 문자를 반환 / 없으면 null 반환
int search(String)	왼쪽부터 파라미터의 문자를 찾아 최초로 일치하는 인덱스 번호 반환
String substr(int, int)	첫번째 파라미터 위치에서부터 두번째 파라미터 갯수만큼 문자열을 반환
Array split(String)	파라미터의 문자를 기준으로 문자데이터를 나누어 배열에 저장해서 반환
String concat(String)	문자열에 파라미터로 입력한 문자열을 더해서 반환
String trim()	문자 앞 뒤의 공백 문자를 삭제해서 반환
int charCodeAt(char)	찾을 문자의 아스키코드값을 반환
char fromCharCode(int)	파라미터(아스키코드 값)에 해당하는 문자를 반환