INTERACTION PROGRAMMING 1

인터랙션 프로그래밍 1

5 Week.

2019. 4. 4.

Array

Array 객체

한 번에 두가지 이상의 값을 포함할 수 있는 객체

사용빈도가 아주 높다.

Array 배열

```
var a = 10;
var b = 'apple';
var c = null;
var d = a;
var _array = [a, b, c, d];
console.log(_array[3]);
```

```
function getMembers(){
    return ['rh', 'june', 'mind'];
}
var members = getMembers();
console.log(members[0]);
console.log(members[1]);
console.log(members[2]);
```

배열의 추가 / 제거 / 정렬

unshift(); push(); shift(); pop();

concat(); splice(); sort(); reverse();

```
var _heros = ['Iron Man', 'Hulk', 'Thor', 'Doctor Strange'];
_heros.unshift('Captain America');
_heros.push('Spider-Man');
_heros.concat(['Black Panther', 'Ant-Man']);
_heros.splice(2, 0, 'Vision');
_heros.splice(2, 1, 'Loki');
_heros.shift();
_heros.pop();
```

```
_array.splice(start, deleteCount, string[]);
```

Array 배열의 제어

```
var _heros = ['Iron Man', 'Hulk', 'Thor', 'Doctor Strange'];
_heros.sort();
_heros.reverse();
```

배열, 객체의 반복문

for ... in

```
function getMembers(){
    return ['rh', 'june', 'mind'];
}
var members = getMembers();
for(var i = 0; i < members.length; i++){
    console.log(members[i]);
}
```

Array 배열의 반복

```
var _heros = ['Iron Man', 'Hulk', 'Thor', 'Doctor Strange'];
for(var name in _heros){
    console.log(name);
}
```

```
var _person = {
    name    : '김용원',
    job     : '교수',
    phone   : '010-9137-8688',
    email   : 'rh@102labs.com'
};
for(var key in _person){
    console.log(key + ' : ' + _person[key]);
}
```

Math

Math 수학

```
Math.abs(-100 + 50);

Math.max(0, 10);
Math.min(0, -10);

Math.max.apply(null, [0, 2, 3, 4, 5]);
Math.min.apply(null, [-5, -4, -3, -2, -1, 0]);
```

Apply

Apply

```
var fruit = { apple : 1000, orange : 2000, lemon : 3000, mango : 4000};
function sum(){
    var result = 0;
    for(var key in this){
        result += this[key];
    }
    return result;
}
console.log(sum.apply(fruit));
```

BOM

BOM (Browser Object Model)

- window
- navigator
- screen
- history
- location

window

window

BOM(DOM Level 0) 의 최상위 객체

브라우저 객체의 최상위 객체

브라우저 환경의 모든 객체 속성을 담고 있다.

document

BOM(DOM Level 0) 의 최상위 객체

브라우저 객체의 최상위 객체

브라우저 환경의 모든 객체 속성을 담고 있다.

브라우저 문서(html)를 모두 읽어들이면 해당 문서를 객체화하여 **document** 객체가 된다.

html 문서의 root 객체로 모든 node 를 갖게 된다.

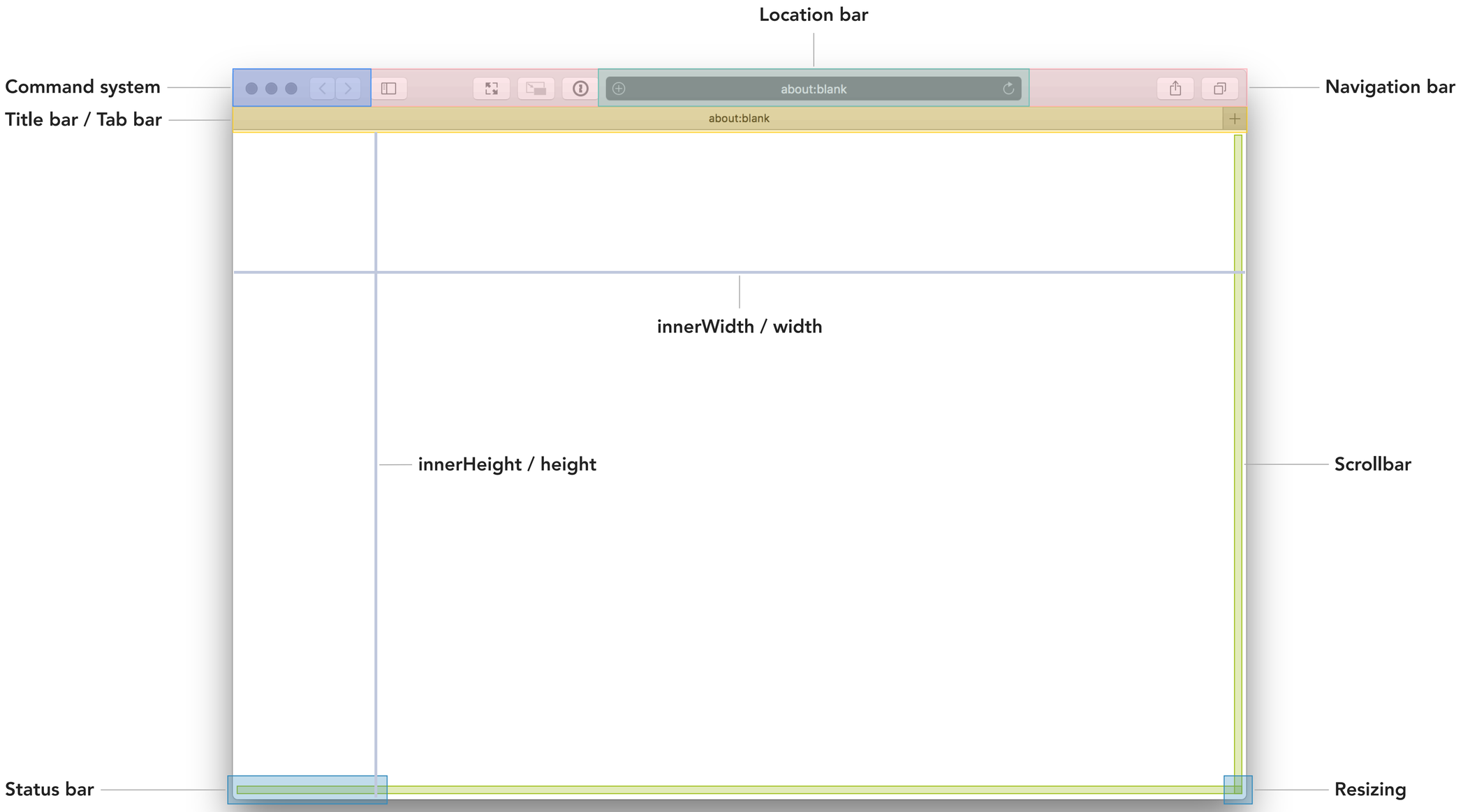
window.innerWidth

윈도우 콘텐츠 영역의 넓이 값(pixel)을 반환한다.

window.innerHeight

윈도우 콘텐츠 영역의 높이 값(pixel)을 반환한다.

window


Location bar
Command system
Navigation bar
Title bar / Tab bar
about:blank
about:blank
innerWidth / width
innerHeight / height
Scrollbar
Status bar
Resizing

window.screenTop

모니터상에서 종(수직) 좌표 값(pixel)을 반환한다.

window.screenLeft

모니터상에서 횡(수평) 좌표 값(pixel)을 반환한다.

window.pageXOffset

문서의 횡(수평) 스크롤 좌표 값(pixel)을 반환한다.

window.pageYOffset

모니터상에서 종(수직) 좌표 값(pixel)을 반환한다.

window.location

현재 웹페이지의 URL 을 반환한다.

window.alert();

window.alert('message');

OK 버튼을 포함하고 있는 경고창을 출력한다.

window.prompt();

window.prompt('message');

메세지와 사용자가 입력 가능한 프롬프트를 포함한 다이얼로그를 출력한다.

window.confirm();

window.confirm('message');

메세지와 '확인', '취소' 버튼을 포함하고 있는 다이얼로그를 출력한다.

window.scrollTo(x, y);

window.scrollTo(0, 0);

문서를 입력한 위치(x, y) 좌표로 스크롤한다.

window.setInterval();

window.setInterval(callback, time);

주기적으로 실행되는 타이머를 생성한다. (ms - millseconds 단위)

window.clearInterval();

window.clearInterval(timer);

setInterval 로 생성된 타이머를 초기화한다.

window.setTimeout();

window.setTimeout(callback, time);

한번만 실행되는 타이머를 생성한다. (ms - millseconds 단위)

window.clearTimeout();

window.clearInterval(timer);

setTimeout 으로 생성된 타이머를 초기화한다.

window.open();

window.**open(URL, name, specs, replace);**

새로운 브라우저 window 를 연다.

URL	새 window 의 URL
name	window 의 이름 (_blank, _parent, _self, _top, name)
specs	선택적인 매개변수
replace	현재 방문이력의 목록에서 대체 여부

window.open();

specs

top=pixels left=pixels width=pixels height=pixels

titlebar=yes|no|1|0 menubar=yes|no|1|0 scrollbars=yes|no|1|0 toolbar=yes|no|1|0

location=yes|no|1|0 status=yes|no|1|0 resizable=yes|no|1|0 fullscreen=yes|no|1|0

channelmode=yes|no|1|0 directories=yes|no|1|0

window.close();

window.close();

target.close();

새로 열린 window 를 닫는다.