

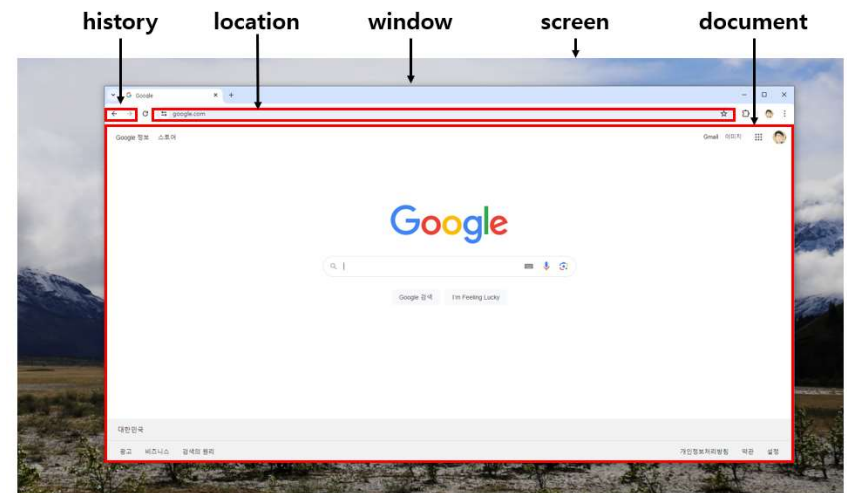
03

03-2. 브라우저 내장 객체

Implicit Object

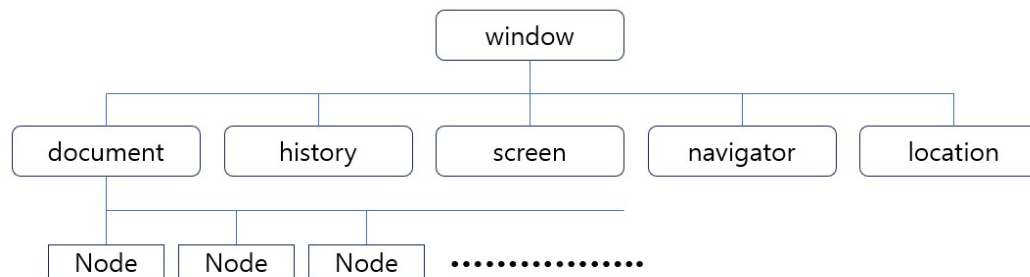
# 브라우저 내장 객체

- 브라우저 내장 객체는 브라우저에서 제공하는 자바스크립트 내장 객체임으로 브라우저에서 실행되는 프론트 웹 애플리케이션에서만 사용할 수 있습니다.
- window : 브라우저 창을 지칭하는 객체
- document : 브라우저에 출력되는 HTML 문서를 지칭하는 객체
- history : 브라우저의 인터넷 방문 기록을 지칭하는 객체
- screen : 브라우저가 떠 있는 스크린 창을 지칭하는 객체
- navigator : 브라우저에 대한 정보를 제공하는 객체
- location : 브라우저의 출력 URL 을 지칭하는 객체



# 브라우저 내장 객체

- 브라우저의 객체들은 모두 window 의 객체들로 window 부터 계층 구조로 구성되어 있습니다.



- document, history, screen, navigator, location 모두 window 에 포함된 객체들입니다.
- 그럼으로 이 객체들을 이용할 때 window.document 혹은 window.history 등으로 이용해야 합니다.
- 브라우저에서 실행되는 모든 자바스크립트의 글로벌 객체가 window 이며 이 window 객체는 생략할 수 있습니다.
- 즉 window.document 로 작성해야 하지만 window 를 생략하고 document 만 작성해도 됩니다.

브라우저 내장 객체 이용

```
1 window.document.querySelector('#a1')
2 document.querySelector('#a1')
```

# window

---

- window 객체는 브라우저 창을 지칭하는 객체로 브라우저에서 실행되는 애플리케이션 모든 곳에서 이용이 가능한 객체입니다.
- alert() 함수도 window 객체의 함수입니다. console 도 window 에 선언된 객체입니다. document, location, navigator, history, screen 도 window 에 선언된 객체입니다.

window 객체 이용	
1	<code>window.alert('hello')</code>
2	<code>window.console.log('hello')</code>
3	
4	<code>alert('world')</code>
5	<code>console.log('world')</code>

# window

---

## window 프로퍼티

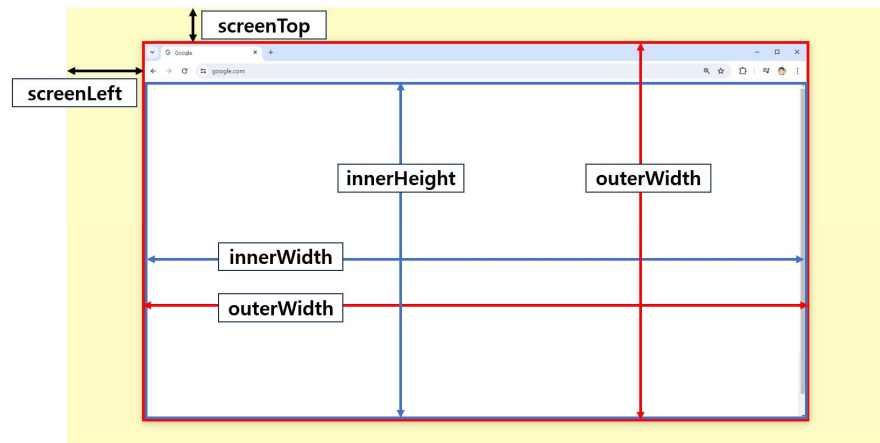
프로퍼티	설명
document	HTML 문서를 지칭하는 객체
location	브라우저의 URL 정보를 지칭하는 객체
screen	스크린 창을 지칭하는 객체
history	브라우저의 인터넷 방문 기록을 지칭하는 객체
navigator	브라우저의 다양한 정보를 가지는 객체
console	브라우저 콘솔 창, 주로 로그 출력에 이용
innerWidth	HTML 문서가 출력되는 부분의 가로 사이즈
innerHeight	HTML 문서가 출력되는 부분의 세로 사이즈
outerWidth	브라우저 창의 가로 사이즈
outerHeight	브라우저 창의 세로 사이즈
screenLeft	스크린 왼쪽에서 브라우저가 위치한 거리
screenTop	스크린 위에서 브라우저가 위치한 거리
scrollX	브라우저의 가로방향 스크롤 위치
scrollY	브라우저의 세로방향 스크롤 위치

# window

---

브라우저의 사이즈 정보가 필요한 경우 `innerWidth`, `innerHeight` 혹은 `outerWidth`, `outerHeight` 를 이용합니다. `outerWidth`, `outerHeight` 는 브라우저 창 자체의 사이즈이며 `innerWidth`, `innerHeight` 는 스크롤 바까지 포함한 HTML 문서가 출력되는 부분의 사이즈입니다.

또한 브라우저가 스크린에서 어느 위치에 보이고 있는지를 `screenTop`, `screenLeft` 를 이용해 알아볼 수 있습니다.



# window

---

## window 함수

함수	설명
alert()	알림 다이얼로그 띄우기
confirm()	확인, 취소 버튼이 있는 다이얼로그 띄우기
prompt()	사용자에게 입력받는 다이얼로그 띄우기
open()	새로운 URL 문서를 띄운다.
close()	브라우저 창 닫기
scrollBy()	스크롤 시키기

# window

---

## window 함수 – open()

- open() 함수를 이용해 새로운 URL 을 출력시킬 수 있는데 이 URL 의 HTML 문서가 새로운 브라우저 창이 열리고 그 창에서 떠나야 하는지 아니면 현재 창에 떠나야 하는지를 지정해 주어야 합니다.
- 브라우저 창에서 새로운 브라우저 창을 띄우면 기존 브라우저 창은 "부모 창(parent)" 이 되며 새로 뜬 브라우저는 "자식 창(child)" 이 됩니다.
- open(url) : url 만 지정해서 open() 함수 호출
- open(url, target) : url 과 target 을 지정해서 open() 함수 호출
- open(url, target, windowFeatures) : url, target 과 다양한 정보를 설정해서 open() 함수 호출



# window

---

## window 함수 – open()

- open() 함수를 호출하면서 target 을 매개변수로 지정해서 새로운 url 이 어느 창이 떠야 하는지를 제어할 수 있습니다.
- target 정보를 지정하지 않으면 기본 \_blank 가 되며 \_blank 는 창을 열고 그곳에 url 을 출력하는 target 정보입니다.
- \_blank : 새 창이 열리고 그곳에 url 출력
- \_parent : 부모 창에 url 출력
- \_self : 자신 창에 url 출력

target 을 _self 로 설정해서 open() 함수 호출
1 window.open("http://www.google.com", "_self")

# window

---

## window 함수 – open()

- open() 함수를 호출하면서 url, target 정보 이외에 windowFeatures 정보를 설정할 수 있는데 windowFeatures 에 창에 대한 정보를 설정할 수 있습니다.
- width, height : 새로운 브라우저 창의 사이즈
- left, top : 새로운 브라우저 창이 뜨는 위치

windowFeatures 정보 설정으로 open() 함수 호출	
1	<code>window.open(</code>
2	<code>    "http://www.naver.com",</code>
3	<code>    "_blank",</code>
4	<code>    "left=100,top=100,width=300,height=400"</code>

# window

---

## window 함수 – close()

- close() 함수를 이용해 브라우저 창을 닫을 수 있습니다.
- 부모창에서 자식창을 open() 띄우면서 자식창을 임의 객체에 저장해 놓고, 그 객체를 이용해 close() 함수를 호출하면 자식창이 닫히게 됩니다.

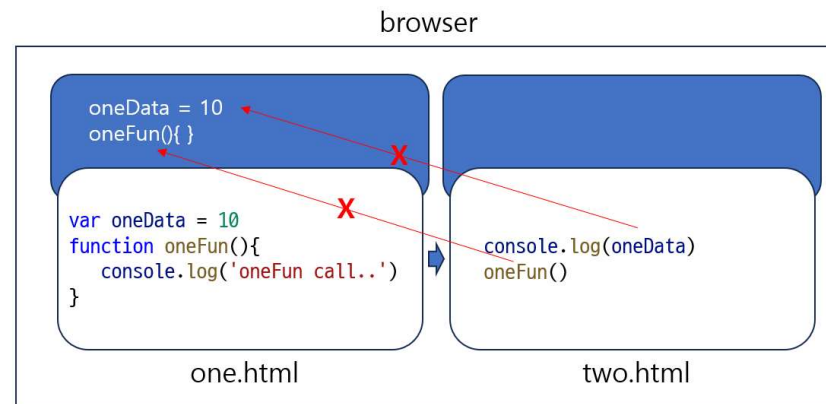
```
childWindow.close()
```

# window

---

## HTML 문서의 메모리 구조

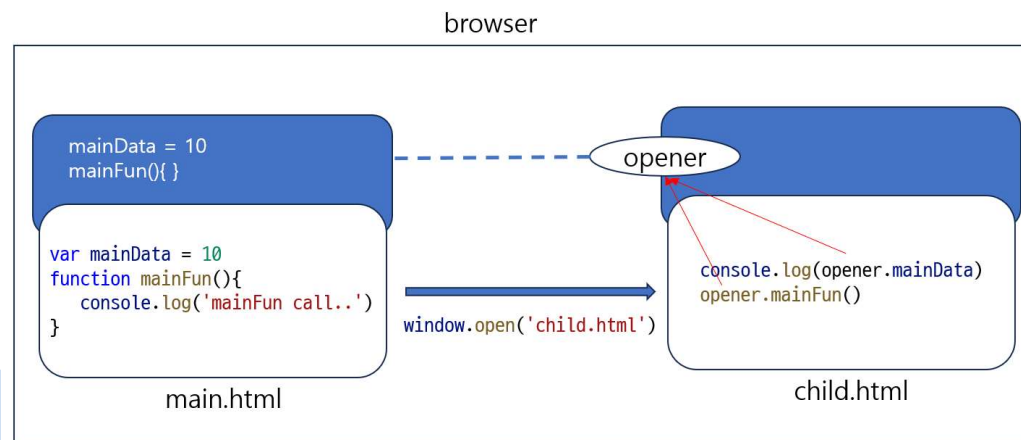
- 자바스크립트에서 변수, 함수등을 이용하기 위해서는 메모리가 할당이 되어야 하고 그 메모리에 변수, 함수 등이 저장되어 이용됩니다.
- 그런데 하나의 HTML 문서를 위해 할당된 메모리를 다른 문서의 자바스크립트에서 접근할 수 없습니다.
- 메모리는 HTML 단위로 할당이 되며 어떤 HTML 문서를 위한 메모리는 그 HTML 문서의 자바스크립트에서만 이용할 수 있습니다.



# window

## HTML 문서의 메모리 구조 - opener 를 이용해 부모창의 변수 함수 이용하기

- open() 함수로 새창을 여는 경우 경우에 따라 자식창에서 부모창의 데이터가 필요할 수도 있고 반대로 부모창에서 자식창의 데이터가 필요할 수도 있습니다.
- open() 함수에 의해 자식창이 열리면 부모창과 자식창이 상호 변수 혹은 함수에 접근해서 하는 경우가 있는데 이를 위해 opener 라는 객체를 제공합니다.
- opener 는 부모창에서 자식창을 open() 함수로 여는 경우 자식창에서 자신을 연 부모창을 지칭하는 객체입니다.



# window

---

## HTML 문서의 메모리 구조 - opener 를 이용해 자식창의 변수 함수 이용하기

- 부모창에서 open() 으로 띄운 자식창에 선언된 변수, 함수가 이용되어야 하는 경우가 있습니다.
- 이것이 가능하려면 먼저 자식창의 자바스크립트 코드에서 opener 객체를 이용해 자신의 window 를 지칭하는 객체를 선언해 주어야 합니다.

```
child.html
1  var childData = 20
2  function childFun(){
3      console.log('childFun call..')
4  }
5  opener.child = this
```

# window

---

## HTML 문서의 메모리 구조 - opener 를 이용해 자식창의 변수 함수 이용하기

- 부모창에서 child 로 자식창에 선언된 변수, 함수를 이용하고 있습니다.
- child 는 자식창에서 opener 에 등록시킨 객체 명입니다.

```
main.html
1  let getChildData = () => {
2      console.log(child.childData)
3      child.childFun()
4  }
```

# window

---

## HTML 문서의 메모리 구조 - opener 를 이용해 부모창의 URL 변경하기

- opener 객체를 이용해 자식창에서 부모창의 URL 을 변경해서 특정 화면으로 이동하게 할 수 있습니다.

child.html	
1	<code>opener.location.href = 'http://www.google.com'</code>
2	<code>window.close()</code>



# location

---

- location 객체는 브라우저의 URL 을 다루기 위한 객체입니다.
- URL 에서 원하는 정보를 추출하는 방법을 제공하거나 브라우저의 화면을 특정 URL 로 변경하는 방법을 제공합니다.

## location 프로퍼티

프로퍼티	설명
href	브라우저의 URL 정보
protocol	URL 의 프로토콜 정보
host	URL 의 호스트 정보, 호스트명과 포트 정보
hostname	URL 의 호스트명정보
port	URL 의 포트 정보
pathname	URL 의 경로 정보
search	URL 의 쿼리 문자열 정보
hash	URL 의 해시 정보
origin	URL 의 백엔드를 식별하기 위한 정보

# location

---

## location 함수

함수	설명
reload()	현재의 URL로 새로 요청
replace()	주어진 URL로 이동

- reload() 함수는 현재의 URL로 다시 요청하는 것임으로 마치 브라우저의 새로고침 버튼을 누른 것과 동일한 효과가 납니다.
- replace() 함수를 이용해 브라우저에 새로운 URL 을 지정할 수 있고 이 URL 로 이동하게 할 수 있습니다.
- location.href 로 이동하면 현재의 URL 정보가 브라우저 히스토리에 유지됨으로 만약 뒤로가기 버튼이 클릭 되면 현재의 화면이 다시 나오게 됩니다.
- location.replace() 로 이동을 하면 현재의 URL 정보는 유지되지 않습니다.

# history

---

- history 객체는 브라우저의 히스토리 정보를 이용해 이전 혹은 이후 화면으로 이동시키기 위해서 사용됩니다.

## history 함수

프로퍼티	설명
back()	이전 화면으로 이동
forward()	이후 화면으로 이동
go()	히스토리의 특정 위치로 이동

- go()함수의 매개변수에 숫자를 지정하면 그 단계의 앞, 뒤로 이동이 가능합니다.
- go( 0 ) 처럼 0을 지정하게 되면 브라우저의 "새로고침" 버튼을 클릭한 것과 동일해 집니다.

# screen

---

- screen 객체는 브라우저가 실행되는 스크린 정보를 획득하기 위해서 이용됩니다.

## screen 프로퍼티

프로퍼티	설명
availWidth	사용 가능한 스크린의 가로 사이즈
availHeight	사용 가능한 스크린의 세로 사이즈
width	스크린 가로 사이즈
height	스크린 세로 사이즈

- availWidth, availHeight 와 width, height 의 차이는 시스템이 기본으로 제공하는 영역까지 포함되는지의 차이입니다.

# navigator

---

- navigator 객체는 브라우저의 다양한 정보 혹은 기능에 접근할 수 있는 프로퍼티를 제공하는 객체입니다.
- navigator 는 화면과 관련 없는 다양한 정보 혹은 기능에 접근할 수 있는데 브라우저의 이름 혹은 브라우저가 네트워크에 연결된 것인지를 navigator 정보로 알 수 있으며 위치정보, 배터리 정보등에도 접근할 수 있습니다.

## navigator 프로퍼티

프로퍼티	설명
cookieEnabled	브라우저에서 쿠키 사용이 가능한지 여부
language	브라우저에 설정된 언어
onLine	브라우저가 네트워크에 연결된 것인지 여부
userAgent	브라우저, 플랫폼 정보

# navigator

---

- userAgent 프로퍼티가 반환하는 문자열을 애플리케이션이 실행되는 브라우저와 그 브라우저가 실행되는 플랫폼 정보입니다.
- 이 정보를 통해 유저가 어느 브라우저에서 혹은 어느 플랫폼에서 애플리케이션을 이용하고 있는지를 확인할 수 있습니다.
- 브라우저는 자동으로 HTTP Request header 에 userAgent 문자열을 전송해 줍니다.
- 그럼으로 백엔드 애플리케이션에서도 확인할 수 있는 정보입니다.



# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare