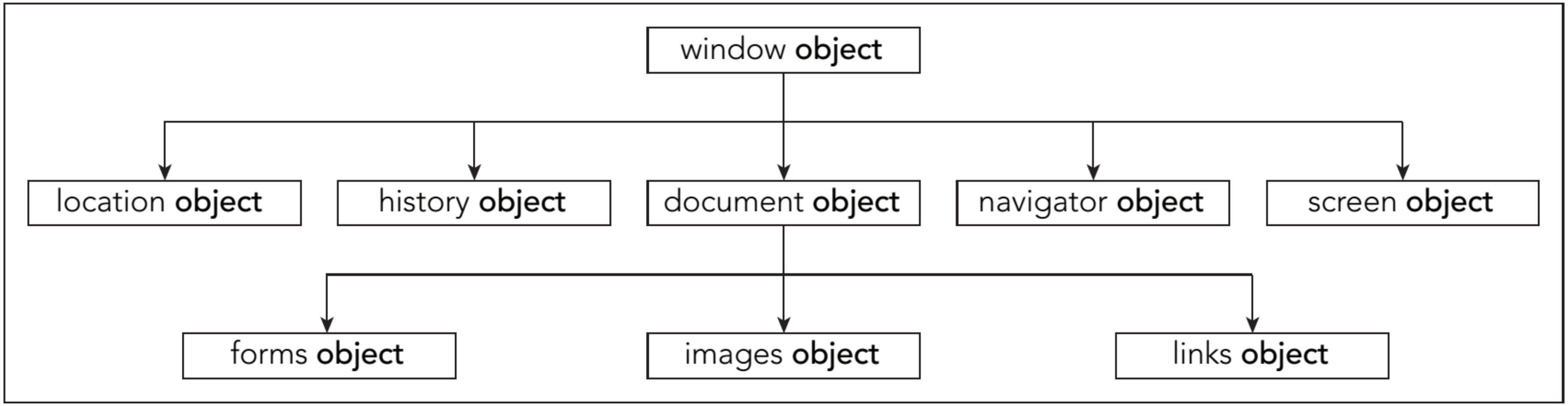
BROWSER OBJECTS

BROWSER OBJECTS

Examples

- window.alert()
- window.prompt()
- document.write()

BROWSER OBJECT MODEL



THE WINDOW OBJECT

- A global object (you don't need to use its name to access its properties and methods).

```
alert("Hello");
window.alert("Hello");
```

Open new window and close it

```
var my_window = window.open("https://www.sait.com");
my_window.close();
```

THE DOCUMENT OBJECT

- one of the most used objects in the BOM
- gain access to HTML elements, their properties, and methods

```
document.bgColor; // get and set the background color of the page
document.images; // get a list of images on the page
```

https://developer.mozilla.org/en-US/docs/Web/API/Document

THE HISTORY OBJECT

- keeps track of each page that user visits
- enables Back and Forward buttons to revisit pages

```
history.length; // how many pages are in the history stack
history.back(); // go back 1 page
history.forward(); // go forward 1 page
history.go(-2); // goes back 2 pages
history.go(3); // goes forward 3 pages
```

THE LOCATION OBJECT

- information about the current page's location
  - URL
  - server hosting page
  - port number
  - protocol

  (need to load a page from a server to see some of these)

```
location.replace("myPage.html"); // removes current page from history stack and replaces it with new page
location.href = "myPage.html"; // goes to new page and adds it to the top of the history stack
```

THE NAVIGATOR OBJECT

- information about the browser and the operating system in which it's running
- often used to handle browser differences because it lets you see browser, version, OS the user has (browser sniffing)
- geolocation

GEOLOCATION

- obtain and use the position of the device or computer

```
function success(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    var altitude = position.coords.altitude;
    var speed = position.coords.speed;
}

navigator.geolocation.getCurrentPosition(success);
```

GEOLOCATION ERROR

- getCurrentPosition() accepts a second parameter.
- use to handle errors

```
function geoError(errorObj) {
    alert("Uh oh, something went wrong");
}

navigator.geolocation.getCurrentPosition(success, geoError);
```

ACTIVITY: GEOLOCATION

- Create a new index.html page.
- Use the geolocation object to retrieve latitude and longitude of the device/computer and write it to the page.
- Create a success function and an error function.

THE SCREEN OBJECT

- contains information about the display capabilities of the client machine

```
screen.height; // height of the screen in pixels
screen.width; // width of the screen in pixels
screen.colorDepth; // number of bits used for colors on client's screen
screen.orientation // orientation of the screen (landscape, portrait)
```

ANIMATING CONTENT

ANIMATING CONTENT

- fade elements in and out
- give elements a swipe animation
- animate them to move around the page

PARTS OF EVERY ANIMATION

- the starting state
- the movement toward the final goal
- the end state; stopping the animation

POSITIONING AND MOVING CONTENT

- In addition to changing the styling, we can also move it!

```
var divAdvert = document.getElementById("divAdvert");
divAdvert.style.position = "absolute";
divAdvert.style.left = "100px"; // set the left position
divAdvert.style.top = "100px"; // set the top position
```

TIMERS

- `setTimeout()` - one-shot timer
- `setInterval()` - continually firing timer

ONE-SHOT TIMER

```
var timerId = setTimeout(yourFunction, millisecondsDelay);
```

SETTIMEOUT

```
function doThisLater() {
    alert("Time's up!");
}

setTimeout(doThisLater, 3000);
```

STOP A TIMER

```
function doThisLater() {
    alert("Time's up!");
}

var timerId = setTimeout(doThisLater, 3000);

clearTimeout(timerId);
```

INTERVALS

```
var myTimerID = setInterval(myFunction, 5000);

clearInterval(myTimerID);
```

ACTIVITY: MAKE A CLOCK

- Create a simple HTML page
- Create a function that displays the current date and time.
- Use setInterval() to call the function every second.

ANIMATION EXAMPLE

Link

ANIMATION EXAMPLE

```
<body>
    <img id="cat" src="img/cat.gif">
    <!-- ...more code -->
```

In CSS:

```
#cat {
    position: absolute;
    left: 0;
}
```

Link

ANIMATION EXAMPLE

```
<script>
    var cat = document.getElementById('cat');
    var catTimer = setInterval(catWalk, 5);

    function catWalk() {
        cat.style.left = cat.offsetLeft + 1 + "px";
    }
</script>
```

ANIMATION EXAMPLE

```
<script>
    var cat = document.getElementById('cat');
    var catTimer = setInterval(catWalk, 5);
    var walkForwards = true;

    function catWalk() {
        if (cat.offsetLeft >= document.body.offsetWidth - cat.offsetWidth) {
            walkForwards = false;
        }

        if (cat.offsetLeft <= 0) {
            walkForwards = true;
        }

        if (walkForwards) {
            cat.style.left = cat.offsetLeft + 1 + "px";
        } else {
            cat.style.left = cat.offsetLeft - 1 + "px";
        }

    }
</script>
```

ANIMATION EXAMPLE

```
<script>
    var cat = document.getElementById('cat');
    var catTimer = setInterval(catWalk, 5);
    var walkForwards = true;

    function catWalk() {
        if (cat.offsetLeft >= document.body.offsetWidth - cat.offsetWidth) {
            walkForwards = false;
        }

        if (cat.offsetLeft <= 0) {
            walkForwards = true;
        }

        if (walkForwards) {
            cat.style.transform = "scaleX(1)";
            cat.style.left = cat.offsetLeft + 1 + "px";
        } else {
            cat.style.transform = "scaleX(-1)";
            cat.style.left = cat.offsetLeft - 1 + "px";
        }

    }
</script>
```

ACTIVITY: CAT WALK

- Recreate the Cat Walk animation
- Create a new index.html page and script.js page and get your cat to walk from the left side to the right side of the page.

Bonus: Move the cat up and down the page too!

Link

COOKIES

COOKIES

As web developers, we can:

- store small amounts of information in a special place on the user's local disk using a cookie
- use these cookies to access data about the user past the first visit

LIMITATIONS OF COOKIES

- You must use document.cookie to write and read cookies. Reading a cookie takes a lot of code.
- The browser limits the amount of cookies it stores and the size they can be.
- Cookies are shared between both the browser and the server so if your server needs a lot of cookies, that leaves you with little to work with.
- Cookies can expire.

WEB STORAGE

WEB STORAGE

- Solves the problems cookies have
- 2 components
  - Session Storage
  - Local Storage
- Stays within the browser and is never transmitted to the server. Storage for JS developers.
- Provides more storage space than cookies.
- Never expires. Remains until you or the user deletes it.

LOCAL STORAGE

- Data stored in key/value pairs.
- Use the localStorage object to set, get, and remove data.

SETTING DATA

```
localStorage.setItem("username", "Janessa");
```

```
localStorage.userName = "Janessa";
```

Why use setItem?

```
localStorage.user name = "Janessa"; // invalid
localStorage.setItem("user name", "Janessa"); //valid
```

GETTING DATA

```
var name = localStorage.getItem("userName");
```

```
var name = localStorage.userName;
```

REMOVING DATA

Remove a key

```
localStorage.removeItem("userName");
```

```
localStorage.userName = null;
```

Remove all keys and values

```
localStorage.clear();
```

IMPORTANT NOTES

- Web storage only stores strings. Keys and their values must be strings.
- Anything that isn't a string will be converted into a string (numbers or objects).

```
localStorage.age = 35;
var age = localStorage.age;
typeof age; // string
```

```
var janeDoe = {
  firstName: "Jane",
  lastName: "Doe",
  age: 35
};

localStorage.person = janeDoe; // nope!
localStorage.person = JSON.stringify(janeDoe); // YEP! Serialize the object.

var savedPerson = JSON.parse(localStorage.person); // Deserialize it.
```

ACTIVITY: CAT WALK, PART 2

- Modify your cat walk code so that it uses localStorage to store the current location of the cat (consider creating a new variable named `currentLeft` to track this).
- When the page loads, check if the information is stored in localStorage, and if so, set the cat to that location.
- Now modify it to remember the direction the cat is walking in (`walkForwards`), and remember that upon page load.

Remember: localStorage stores strings.