

OIM3690 - Web Technologies



Animation using JavaScript

setTimeout()

- `setTimeout()` is a built-in function in JavaScript that allows you to schedule the execution of a function or a block of code after a set amount of time.
 - It is useful when you need to delay the execution of a piece of code, such as an animation or a network request, to avoid blocking the main thread.
 - It will execute **only once** after the specified delay.
- Syntax:

```
setTimeout(function, delay);
```

setTimeout() Example

- Example:

```
setTimeout(function () {  
    console.log("See me in 1 second!");  
}, 1000);
```

- This example schedules the execution of an anonymous function that logs a message to the console after a delay of 1000 milliseconds (i.e., 1 second).
- Note: `setInterval()` is another built-in function that is similar to `setTimeout()`.
 - However, `setInterval()` executes the specified function repeatedly at a set interval until stopped or cleared.
 - This can be useful when you need to repeatedly execute a task, such as polling a server for updates.

setTimeout() in Recursion

- Example:

```
function animate() {  
  // Some animation code here  
  setTimeout(animate, 1000); // Recursive call  
}
```

- Function `animate()` contains some animation code, followed by a call to `setTimeout()` with a delay of 1000 milliseconds and a **recursive** call to **itself**.
- As a result, the animation code is executed repeatedly, creating an animation loop that is scheduled to run every 1000 milliseconds.
- **Recursion** is a powerful programming technique that can be used to simplify complex problems, but it can also lead to infinite loops if not used properly.
 - Try: Google "[recursion](#)"

Global Variables vs. Local Variables

- **Global** variables:
 - Variables declared **outside** of a given function.
 - They are accessible throughout the whole code, including within functions.
 - They can be used to maintain values that should not get reset during the execution of a program.
- **Local** variables:
 - Variables declared **inside** a given function.
 - They can only be accessed within that function.
 - They are created every time the function is called and destroyed when the function returns.

Examples of Global Variables vs. Local Variables

- Example 1 (**local** variable `counter`):

```
function doThis() {  
  let counter = 0;  
  counter++;  
}
```

- Every time `doThis()` is called, `counter` is set to 0 and then incremented to 1.

- Example 2 (**global** variable `counter`):

```
let counter = 0;  
function doThat() {  
  counter++;  
}
```

- Every time `doThat()` is called, `counter` will **NOT** be reset to 0.
- `counter` is incremented based on its previous value.

Exercise: Creating a Slide-show

- Create a new HTML file, **ex21.html**.
- Declare a global variable `slides` as an array of image sources:

```
const slides = [  
  "images/tiger1.jpg",  
  "images/tiger2.jpg",  
];
```

- Add an `` tag to your HTML code as a placeholder for the changing image.
- Use event `load` on `body` (or `DOMContentLoaded` on `document` - see [difference](#)) to start automatically.
- Write a function, `changeSlide()` to **change** the image.
- Use `setTimeout()` to **recursively** call the function every 1000 milliseconds.
- Code is in the next slide.

ex21.html

```
const slides = [  
  "images/tiger1.jpg",  
  "images/tiger2.jpg",  
  // more  
];  
let currentIndex = 0;  
  
function changeSlide() {  
  const tigerImage = document.getElementById("tiger");  
  if (currentIndex === slides.length) {  
    currentIndex = 0;  
  }  
  tigerImage.src = slides[currentIndex];  
  currentIndex++;  
  setTimeout(changeSlide, 1000);  
}  
  
document.addEventListener("DOMContentLoaded", function () {  
  changeSlide();  
});
```

Exercise: *ex21.html* (cont.)

- Can you add `border` that changes color along with the changing image?
- Can you add **changing text** that describes the changing image?
- Update *sitemap.html* and **commit/push** to GitHub.

Another Animation using JavaScript - Moving Lyrics

- Download [lec21-js-moving-lyrics.html](#) from *OIM3690/resources/templates*
- Read code, including
 - JavaScript
 - CSS (**important** in this example)
- Answer the following questions:
 - What are the **global** variables?
 - What is the **purpose** of each variable?
 - Can you make the movement **faster? More smoothly?**

Questions?

