



# MIS3690 WEB TECHNOLOGIES

**BABSON COLLEGE**  
**TOIM DIVISION**



# ANIMATION WITH JAVASCRIPT



# KEY CONCEPTS IN ANIMATION

- Array variable
- We can declare an array variable as:
  - `let colors= new Array("green", "yellow", "blue", "pink", "red", "orange");`
- Random number generator
  - `Math.random()`

# KEY CONCEPTS IN ANIMATION

- `setTimeout()` – function
  - Allows you to call and execute the same function over and over again, in preset time intervals
  - E.g., `setTimeout("animate()", 1000)`
  - The function `animate()` is called over and over every 1000 micro-seconds, i.e., 1 second)
- It is placed inside the same function that it calls
  - `setTimeout("animate()", 1000)` is placed inside the `"animate()"` function!
  - Known as a **"recursive"** call.

# KEY CONCEPTS IN ANIMATION

## ■ Global variables

- Variables that are declared "OUTSIDE" a given function.

- Used to maintain values that should not get reset.

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

- Every time we call `doThis()`, the counter will be set to 0 and then increased by 1.

- Every time we call `doThat()`, the counter will NOT BE RESET TO 0, it will be incremented by 1, based on its previous value.

# KEY CONCEPTS IN ANIMATION

- Array variable and preloading
- We can declare an array variable as:
  - `const slides = new Array ("tiger1.jpg", "tiger2.jpg");`
- We have an array variable that contains two images in its first 2 index positions.
- Declaring the array variable as a "global" variable and pre-loading the content (images, text, etc.) makes the animation faster (web page loads and works faster).

# CREATING A SLIDE-SHOW

- Set up a place holder for the image using the `<img>` tag in HTML.
- Use the `onload` event in the `<body>` tag to start the animation (typical implementation).
- Define the global array variable to hold the images in the slideshow
- Write the function to change the image
- Use `setTimeout()` to recursively call the function.
- Use CS17-InClass1.htm to implement this using the four tiger images (also provided).

# CREATING A SLIDE-SHOW – THE FUNCTION

```
const slides = ["tiger1.jpg", "tiger2.jpg", "tiger3.jpg",  
"tiger4.jpg"];  
let i = 0;  
function slideShow() {  
    let img = document.getElementById("tigerImage");  
    if (i == slides.length) {  
        i = 0;  
    }  
    img.src = slides[i];  
    i++;  
    setTimeout("slideShow()", 1000);  
}
```



# EXTENDING THE SLIDE-SHOW

- Can we add a border that changes color with the changing image?
- Add another array variable for the colors
- Add another variable to act as the "counter" for colors.
- Can you modify the function to change border colors?
- When done, save it as CSI7-InClass1.htm.

# CHANGING MESSAGES

- Same concept as the slide show.
- Substitute message strings for images
- Use "`innerHTML`" to change the message on the web page!

# CHANGING MESSAGES – THE FUNCTION

```
let counter=0;

const messageList = ["Hey Jude, don't make it bad", "Take a sad song
and make it better", "remember to let her into your heart", "Then you
can start to make it better"];

function messUp()
{
    document.getElementById("msg").innerHTML = messageList[counter];
    counter++;
    if (counter == messageList.length)
        counter = 0;
    setTimeout("messUp()", 2000);
}
```

# SCROLLING TEXT

- Same concept, except we now have to move the text from right to left like a marquee
- We need a place holder for the message line – it must have an `id`
- We need to decide what our starting point is and where we should end the movement.
- Let us assume that we start at `300px` from the left edge and we stop when the text has moved to the `700th` pixel from the left edge.
- We will use the `"onload"` event to load our function.
- We will call the function `"doScroll()"`
- We will use an array to preload the message lines
- We will use a variable to keep track of the position and another variable for our counter

```
let p = 300;
let k = 0;
const messageList = ["Hey Jude, don't make it bad", "Take a sad song and
make it better", "remember to let her into your heart", "Then you can start
to make it better"];

function doScroll()
{
    document.getElementById("msg").style.left = p + "px";
    if ( p > 700)
    {
        p = 300;
        k++;
        if ( k == messageList.length)
        {
            k = 0;
        }
        document.getElementById("msg").innerHTML = messageList[k];
        document.getElementById("msg").style.left = p + "px";
    }
    else
    {
        p = p + 5;
    }
    setTimeout("doScroll()", 100);
}
```

# SCROLLING TEXT

- The most important piece here is the positioning.
- Use CSS to position the text using the "absolute" positioning command.
- This is IMPORTANT, because, without this, moving the text will not work!