introduction to media computing
week 09

School of creative media, mw

Today’s topics (week 09)

- quick review of Arrays and its related functions.

Today's topics (week 09)

- quick review of Arrays and its related functions.

- Useful math functions
- Resource preparation with `preload()`
- Image and Font resources

Review: Array

```
let activeColor = 0;
let color0 = 255;
let color1 = 200;
let color2 = 150;

if (activeColor == 0) {
  fill(color0);
}
else if (activeColor == 1) {
  fill(color1);
}
else if (activeColor == 2) {
  fill(color2);
}
```

```
let activeColor = 0;
let colors = [];
colors[0] = 255;
colors[1] = 200;
colors[2] = 150;

fill( colors[activeColor] );
```

When `activeColor` equals to 1, then we are filling with `colors[1]`.

Review: Array related functions

js

Array property

`.length`	returns number of elements

Array functions

`.push(A)`	appends `A` to the array
`.pop()`	removes the last element
`.slice(s,e)`	copies some elements
`.sort()`	sorts alphabetically

Review: `.sort()` numerically*

We have to provide a special 'compare function' to `.sort()` in order to sort numerically. We may use it as-is at this moment.

Example C

```
let numbers = [200,1000,30];
numbers.sort(function(a, b){return a - b});
// numbers = [30,200,1000]
```

Review: `map()`

`map()` maps a given number `s` within a range defined by `sStart` and `sEnd` to a new destination range defined by `dStart` and `dEnd`.

Example: `map(0.5, 0, 1, 0, 200)` returns `100`

Useful math functions

Math functions `round()`, `ceil()` & `floor()`

When we work with 'array', we have to access its members via an index which is an integer. It is quite often that we want to get an integer from a given decimal number.

function	returns	examples
`round(x)`	a rounded number	`round(3.4)-> 3` `round(3.5)-> 4`
`floor(x)`	the largest integer < `x` i.e. floor of `x`	`floor(3.4)-> 3` `floor(3.5)-> 3`
`ceil(x)`	the smallest integer > `x` i.e. ceiling of `x`	`ceil(3.4)-> 4` `ceil(3.5)-> 4`

Using p5.js `floor()`

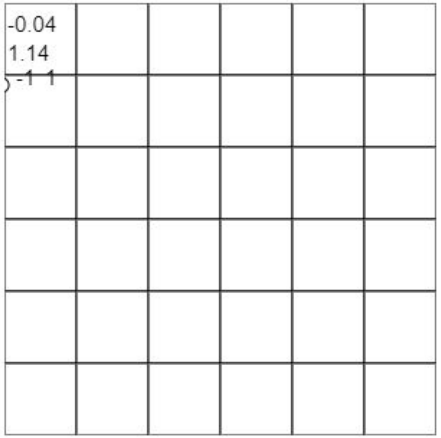
Use simple division and `floor()` to convert mouse cursor position to integer coordinates of the underlying grid.

JS

```
let numDiv = 6; //  Number of divisions
let divSize;    //  Size of each division (pixels)

function setup(){
  createCanvas(300,300);
  textSize(15);
  divSize = width/numDiv;
}

function draw() {

  //  Draw a grid
  for (let y = 0; y < height; y += divSize) {
    for (let x = 0; x < width; x += divSize) {
      rect( x,y, divSize, divSize);
    }
  }
```

Result

EDIT ON CODEPEN



Resources

1x 0.5x 0.25x

Rerun

Using Resources with `preload()`

Resources to be used in a sketch

Similar to a web page, we may use existing images, videos, fonts or audio in our own p5.js sketches; they are collectively known as 'resource'.

Resource type	examples
image	.png, .jpg, .svg, etc.
font	.ttf, .otf, etc.
audio	.mp3, .wav, .aiff, etc.
video	.mov, .avi, etc.
data	.json, .xml, etc.

Pre-loading resources with `preload()`

In order to use the mentioned resources, we need to use a new p5.js function named `preload()`. This function is similar to the `setup()` and `draw()` functions that we have been using, and we have to put all resource loading instructions inside `preload()`.

Pre-loading resources with `preload()`

```
<html>
<head>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.js"></script>
  <script>
  function preload() {
    font01 = loadFont('font01.ttf');
  }
  function setup() {
  }
  function draw() {
  }
  </script>
</head>
</html>
```

Font Resource

`loadFont()` & `textFont()`

Font loading: `loadFont()`

p5.js supports OpenType (.otf) and TrueType (.ttf) fonts. Fonts must be loaded via using `loadFont()` inside the function `preload()` before using them. The font file may be a local file or a URL served by a web server.

```
let font01, font02;
function preload() {
  font01 = loadFont('./fontNumberOne.ttf');
  font02 = loadFont('https://someserver.com/somefont.ttf');
}
```

Font to use: `textFont()`

Once a font has been loaded. You may use the `textFont()` function to define the desired font to be used with the `text()` function.

```
function draw() {
  textFont(font02);
  text( "Hello", 30, 30);
  ...
}
```

Image Resource

`loadImage() & image()`

Image loading: `loadImage()`

p5.js supports major image types. Images should be loaded via using `loadImage()` inside the function `preload()` before using them. The image file may be a local file or a URL served by a web server.

```
let image01, image02;
function preload() {
  image01 = loadImage('./someImage.jpg');
  image02 = loadImage('https://someserver.com/someimage.jpg');
}
```

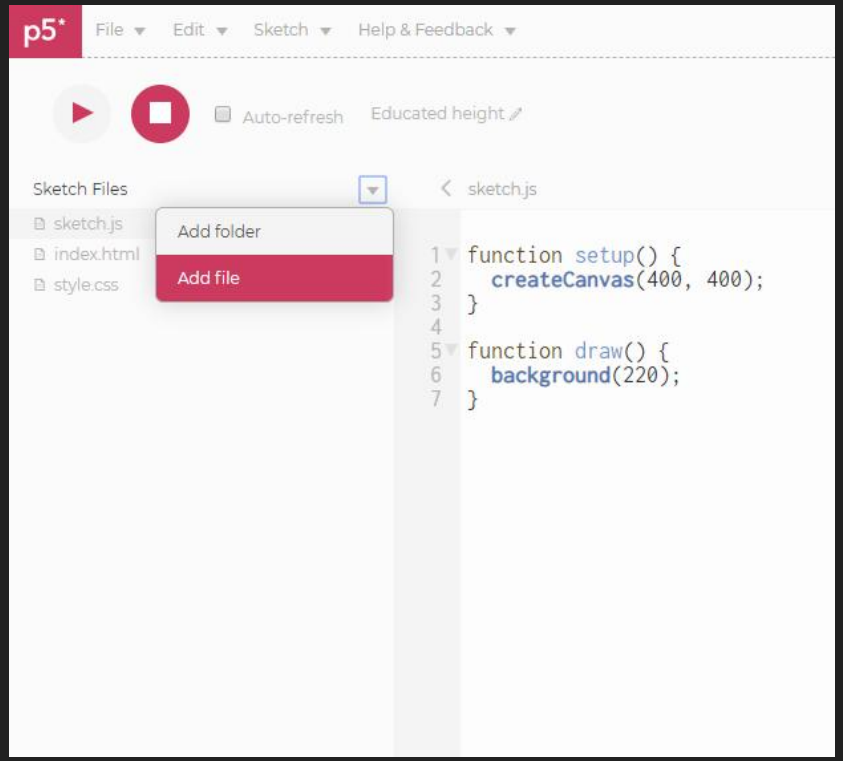
Display image: `image()`

Once the images are pre-loaded using `loadImage()`. We may use `image()` to display them. `image()` supports:

```
image(img,x,y);//(x,y) upper left coord.
image(img,x,y,[width],[height]);
// width,height: scaled width & height on canvas.
```

```
function draw() {
  image( image01, 100, 100 );
  image( image01, 0, 0, 100, 100 );
}
```

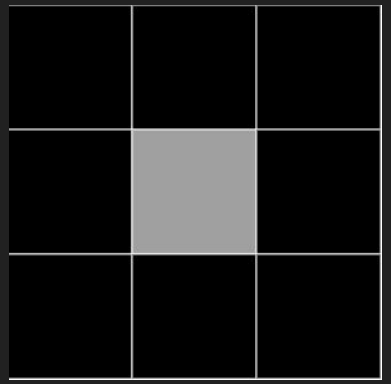
Using resource with p5.js editor

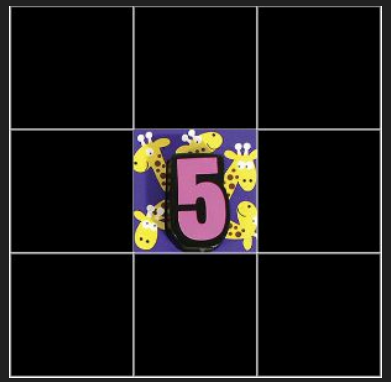


As many web browser will fail to load images or font files locally due to a security policy (CORS). If you use FireFox, you have an option to override it. We strongly recommend using p5.js editor to test the resource related sketches but you need a p5js.org account.

exercise

In-class exercise





Use p5.js editor for this exercise

1. Assume a canvas of size 300 x 300. Use a double for-loop to draw a 3 x 3 grid of black squares of size 100 x 100 each. While the mouse is over a square, that square should change color. (Hint: Apply the `floor()` example to track the mouse, and use an array to store the squares' colors).

2. Instead of changing colors, each square will display a unique image while the mouse is over it. You may use the 9 images here.