introduction to media computing
week 03

School of creative media, mw

Today's topics (week 03)

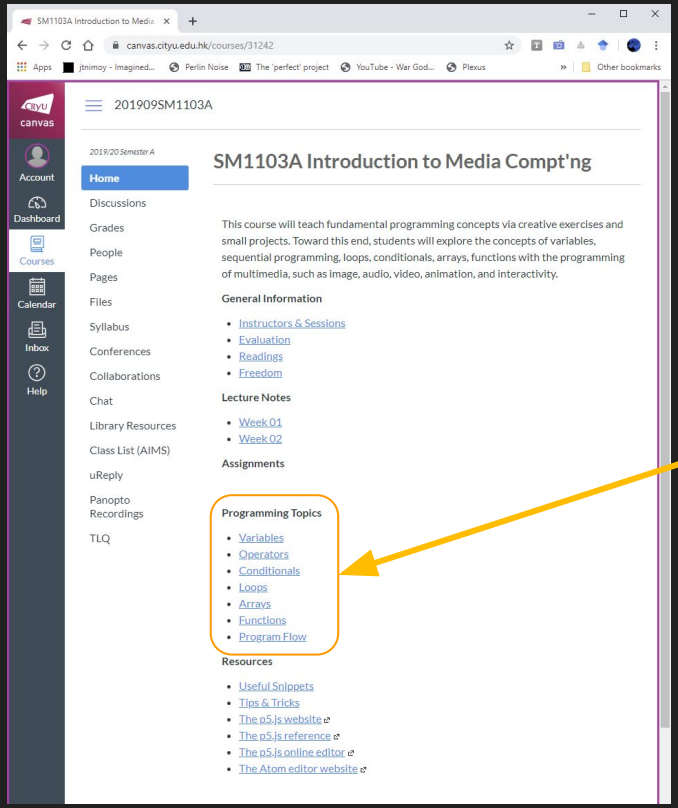
- **operators & conditionals**
  - review
  - the modulo operator `'%'`
- **logical operators**
- **coding style**
- **loops I: `while()` loop**

Today's topics (week 03)

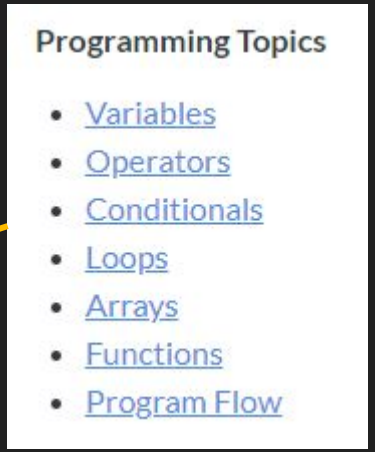
- **operators & conditionals**
  - review
  - the modulo operator `'%'`
- **logical operators**
- **coding style**
- **loops I: `while()` loop**

- **p5.js online editor**
- **keyboard interactivity**
- **drawing text**

Resources for review



https://canvas.cityu.edu.hk/courses/31242



Review: math. operators

```
let a = 10;
let b = 6;
let result;
```

`result = a + b;`	addition	`result = 16`
`result = a - b;`	subtraction	`result = 4`
`result = a * b;`	multiplication	`result = 60`
`result = a / b;`	division	`result = 1.6667`
`result = a % b;`	modulo	`result = 4*`

*Remainder of integer division

Review: assignment operators

```
let r;
r = 10;
r = r + 1;
r += 2;
r -= 2;
r *= 2;
r /= 2;
r %= 2;
```

assignment	r = 11	(10 + 1)
add. assignment	r = 13	(11 + 2)
sub. assignment	r = 11	(13 - 2)
mul. assignment	r = 22	(11 * 2)
div. assignment	r = 11	(22 / 2)
mod. assignment	r = 1	(11 % 2)*

*Remainder of integer division

Review: other operators

```
let x = 200;
x++;
```

`'++'`: increment

`x++` is equivalent to `x = x + 1`

```
let x = 200;
x--;
```

`'--'`: decrement

`x--` is equivalent to `x = x - 1`

Review: if else

```
if (x == 200) {
    //  Do something
}
else if (x < 200) {
    //  Do something
}
else {
    //  Do something else
}
```

Only ONE block of code will be executed.

Review: relational operators

```
if (x >= 200) {
    //  Do something
}
else {
    //  Do something else
}
```

operators	meaning
>	larger than
<	smaller than
>=	larger or equal to
<=	small or equal to
!=	not equal to
==	equal to

Modulo Operator '%'

Modulo Operator '%'

- modulo operator `'%'` computes the remainder of an integer division. Example: `5 % 2` returns 1.

- This operator is particularly useful for some simple looping operation.

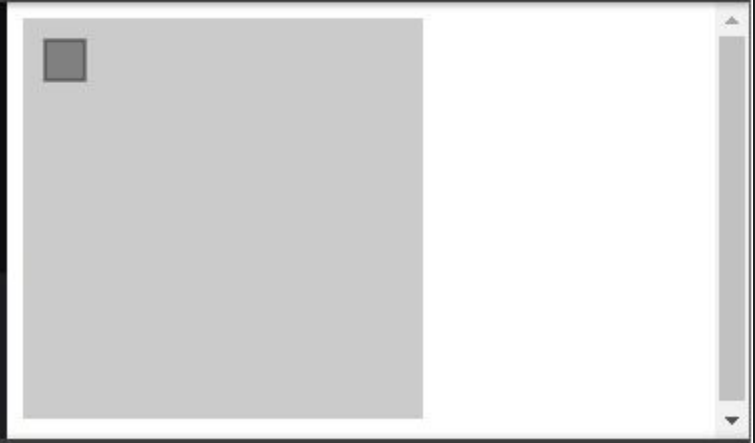
Modulo Operator '%'

JS

```
let num = 0;

function setup(){
  createCanvas(200,200);
  fill(128);
}

function draw() {
  //  Loop thru 0 - 256 for color
  let brightness = num % 256;
  background(brightness);
```

Result

EDIT ON CODEPEN

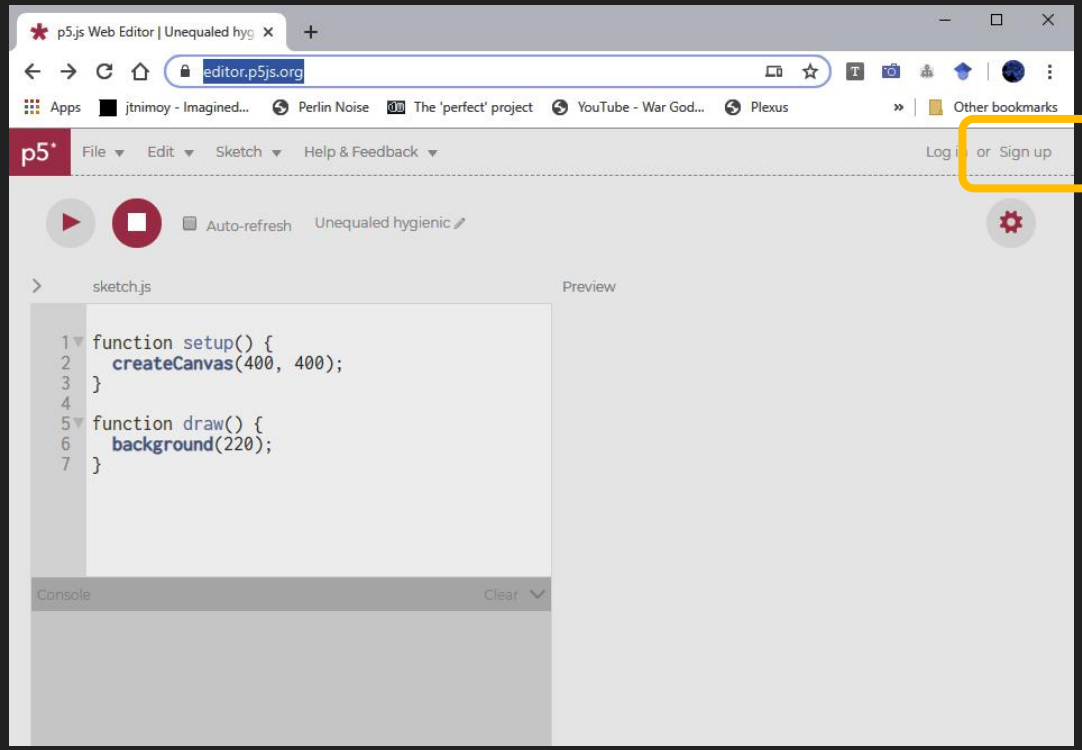


Resources | 1x | 0.5x | 0.25x | Rerun

try https://canvas.cityu.edu.hk/courses/31242/pages/week-03#A

p5.js online editor

p5.js online editor



Please follow the URL and create an account so you can save your sketches, then do the exercise on the next slide.

https://editor.p5js.org/

exercise

In-class exercise 1

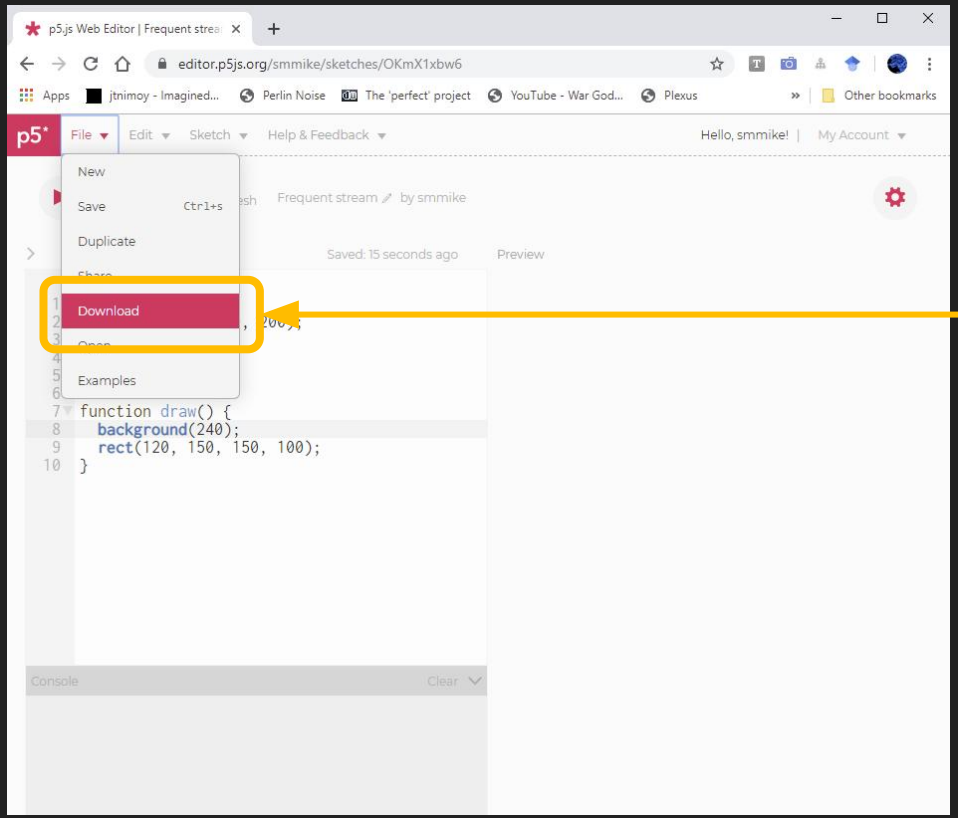
p5*

```
function setup() {
 createCanvas(200, 200);
}
function draw() {
 background(frameCount % 256);
}
```

Modify this program to use ONE conditional and the modulo operator such that the background alternates every 60 frames, between black and white.

Don’t change the frameRate

p5.js online editor



Once you have saved your sketch, you may download the sketch as a zipped archive.

Logical Operators

Logical operators

- logical operator helps us to compose more flexible 'conditions' for various JavaScript conditionals.
- All 'conditions' evaluation in JavaScript returns a logical (boolean) value `'true'` or `'false'`.

```
if (x == 200) {
    //  Do something
}
```

Simple SINGLE condition, what if we want to combine two or more conditions ?

Logical operators

operators	meaning
`||`	Logical OR
`&&`	Logical AND
`!`	Logical NOT

```
if (x == 0 || x == 200 ) {
    //  Do something
}
```

This block will run only if

X equals to 0

OR

X equals to 200

Logical operators

operators	meaning
`||`	Logical OR
`&&`	Logical AND
`!`	Logical NOT

```
if (x > 1 && x != 200) {
    //  Do something
}
```

This block will run only if

X is larger than 1

AND

X is not equal to 200

Logical operators

operators	meaning
`||`	Logical OR
`&&`	Logical AND
`!`	Logical NOT

```
if (!(x > 1)) {
    //  Do something
}
```

The NOT operator always inverses the result of the condition `(x > 1)`. So the block runs when `x` does not fulfill `(x > 1)`, i.e. the block runs when x is NOT larger than `1`.

Truth table

Logical OR '||'

A	B	(A || B)
true	true	true
true	false	true
false	true	true
false	false	false

Logical AND '&&'

A	B	(A && B)
true	true	true
true	false	false
false	true	false
false	false	false

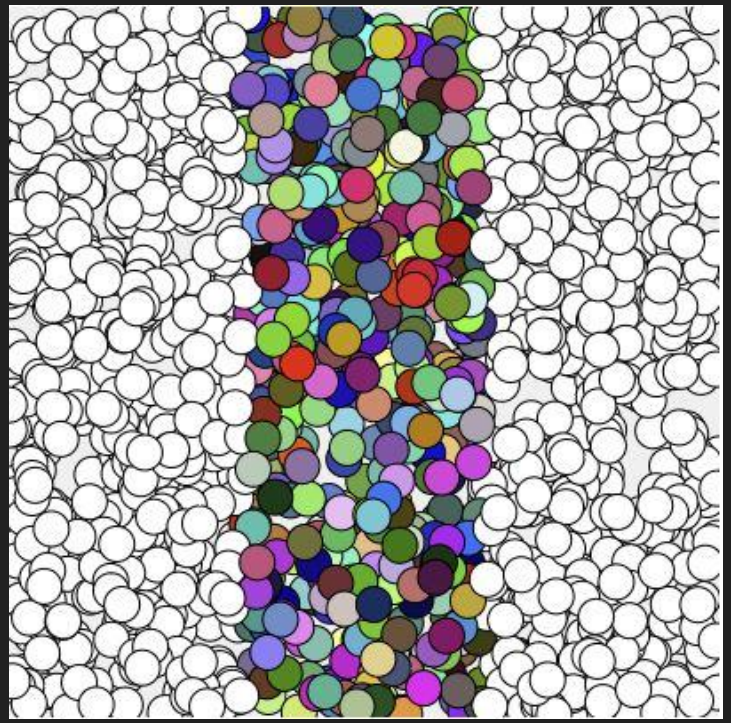
Logical NOT '!'

A	!(A)
true	false
false	true

exercise

In-class exercise 2

p5*



1. Fill the canvas (400 x 400) with circles of size 20, each with a random position.

2. Divide the screen into 3 regions as shown in the figure. Circles in the middle region are randomly colored, and the rest are in white.

3. Use only one conditional. Hint: Use a ' logical operator '.

Coding Style

Coding Style

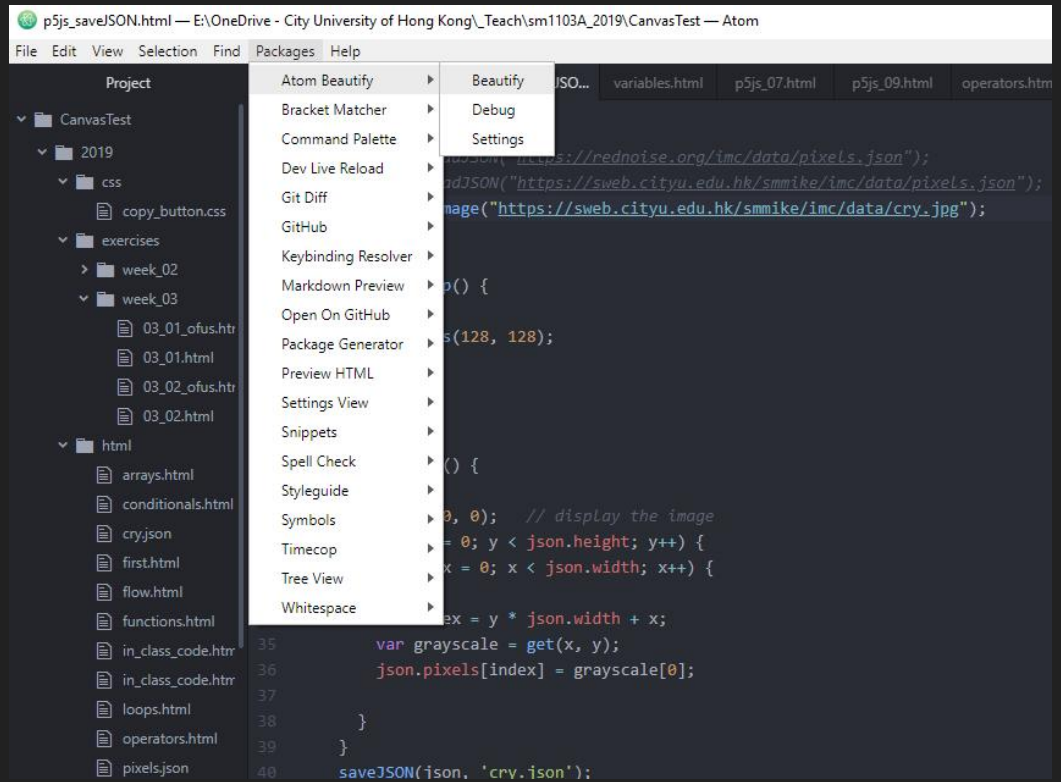
```
let x=0;

function setup() {
  createCanvas(400, 400);
  background(220);
}

function draw() {
  if (x % 2 == 1) {
    fill(150);
  } else {
    fill(0);
  }
  ellipse(50 * x, 0, 50, 200);
  x++;
}
```

It is VERY IMPORTANT to use proper indentation and spacing in your code.

WHY ?

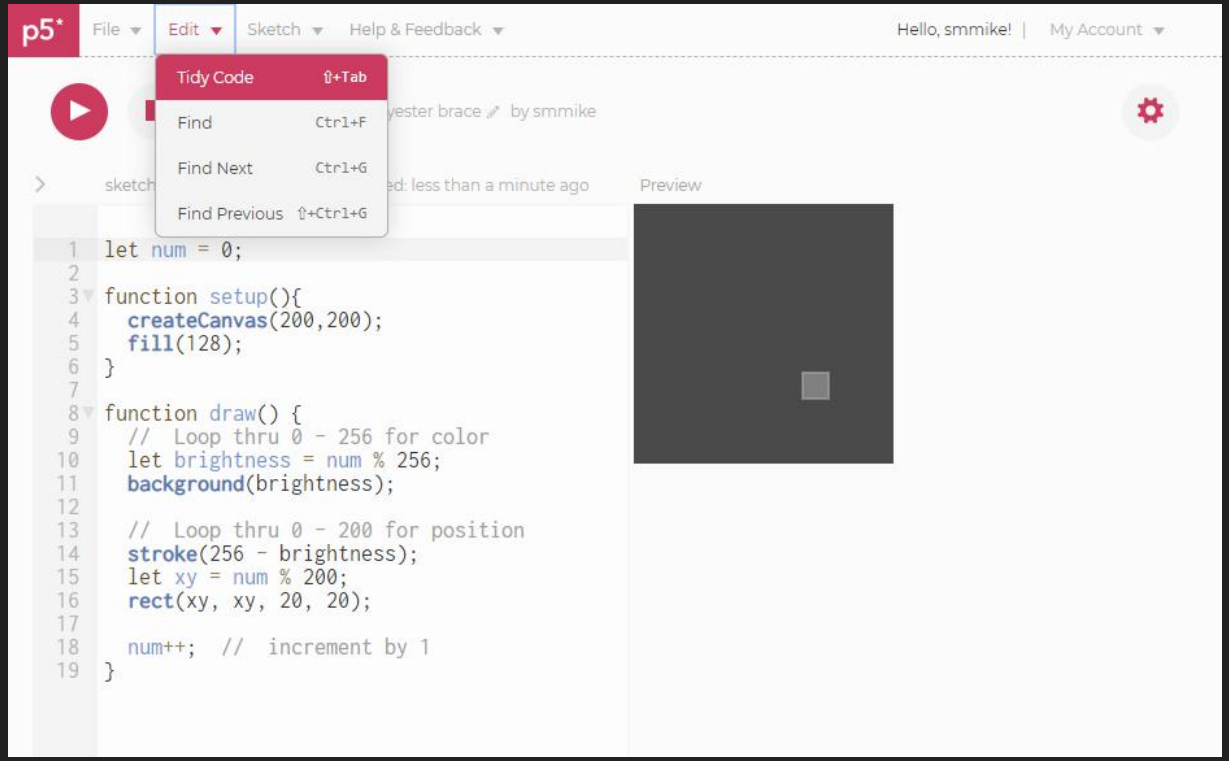
- **Easier for you to read**
- **Easier for others to read**
- **Help you to spot mistakes**
- **It shows you understand *the craft of coding***
-

Coding Style



Most modern code editor like Atom often has code 'beautify' or 'prettify' package which helps with code formatting and syntax highlighting.

Coding Style



p5.js editor also offers convenient code formatting functions.

Coding Style

Properly formatted code is required in all assignments. Poorly formatted code will lead to point deduction.

Loops 1: The 'while()' loop

Loops 1: the `while()` loop

- A 'Loop' allows a block of code to be executed repeatedly (aka iteration).

- A `while()` loop repeats a block of code to as long as certain condition is fulfilled in each iteration.

Loops 1: the `while()` loop