

introduction to media computing week 03



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

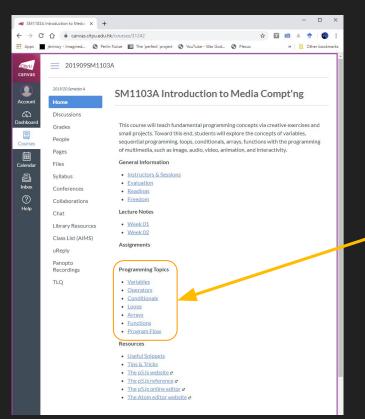


- p5.js online editor
- keyboard interactivity
- drawing text



Resources for review





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

Programming Topics

- Variables
- Operators
- Conditionals
- Loops
- Arrays
- Functions
- Program Flow



js

Review: math. operators

```
let a = 10;
let b = 6;
let result;
result = a + b;
                                         result = 16
                        addition
result = a - b;
                        subtraction
                                         result = 4
result = a * b:
                        multiplication
                                         result = 60
result = a / b;
                        division
                                         result = 1.6667
result = a % b;
                                         result = 4*
                        modulo
                                         *Remainder of integer division
```



Review: assignment operators

```
js
```

```
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

Review: if else

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

Only <u>ONE</u> 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 '%' computes the remainder of an integer division. Example:
 5 % 2 returns 1.

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



Modulo Operator '%'



```
EDIT ON
                                                   Result
                                                                                      CODEPEN
let num = 0;
function setup(){
  createCanvas(200,200);
  fill(128);
function draw() {
  let brightness = num % 256;
  background(brightness);
                                                       0.5× 0.25×
Resources
                                                                                              Rerun
```



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



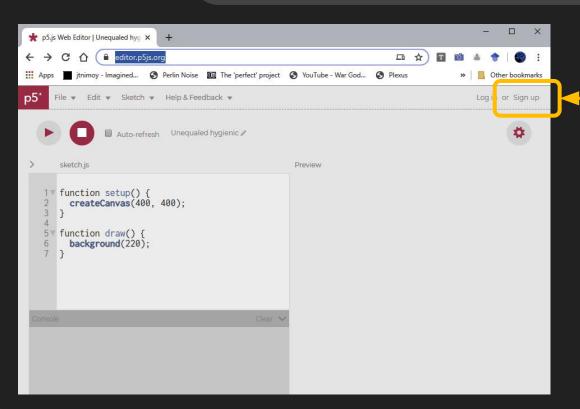


p5.js online editor



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.





In-class exercise 1



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

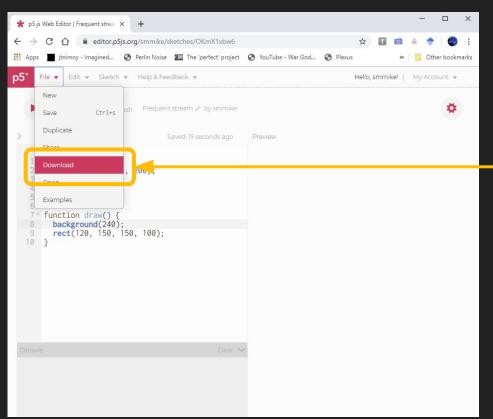
Modify this program to use ONE conditional and the modulo operator such that the background alternates every 60 frames, between black and white.

Don't change the frameRate









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
- 11	Logical OR
&&	Logical AND
1	Logical NOT

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

This block will run only if

X equals to 0

OR

X equals to 200



js

Logical operators

operators	meaning
Ш	Logical OR
&&	Logical AND
į.	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
- 11	Logical OR
&&	Logical AND
į.	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	В	(A B)
true	true	true
true	false	true
false	true true	
false	false	false

Logical AND '&&'

Α	В	(A && B)
true	true	true
true	false	false
false	true false	
false	false	false

Logical NOT '!'

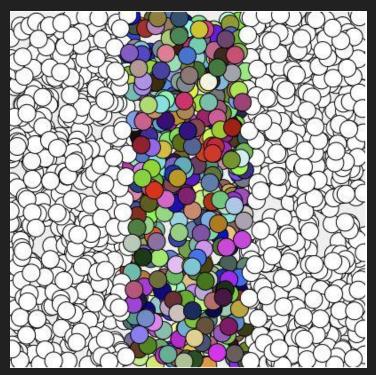
Α	! (A)
true	false
false	true





In-class exercise 2





- 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 <u>only one conditional</u>. Hint: Use a 'logical operator'.









```
function setup() {
 createCanvas(400, 400);
 background(220);
function draw() {
   fill(150);
   fill(0);
 ellipse(50 * x, 0, 50, 200);
```

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





```
@ p5is saveJSON.html — E:\OneDrive - City University of Hong Kong\ Teach\sm1103A 2019\CanvasTest — Atom
File Edit View Selection Find Packages Help
                                  Atom Beautify
                                                          Beautify
            Project
                                  Bracket Matcher
                                                          Debug

✓ ■ CanvasTest

                                  Command Palette
                                                          Settings

→ 2019

                                  Dev Live Reload
      V CSS
                                  Git Diff
                                                        age("https://sweb.cityu.edu.hk/smmike/imc/data/cry.jpg");
          copy button.css
                                  GitHub

✓ 

exercises

                                  Keybinding Resolver
        > week 02
                                  Markdown Preview
                                                      p() {
        ∨ = week 03
                                  Open On GitHub
             (a) 03_01_ofus.htr
                                                      5(128, 128);
                                  Package Generator

    □ 03 01.html

                                  Preview HTML
            (a) 03_02_ofus.htm
                                  Settings View
             (a) 03_02.html
                                  Snippets
      ∨ html
                                  Spell Check
          arrays.html
                                  Stylequide
          conditionals.html
                                  Symbols
          cry.json
                                  Timecop
          first.html
                                                        = 0; x < json.width; x++) {
                                  Tree View
          flow.html
                                  Whitespace
          functions.html
                                              var grayscale = get(x, y);
          in class code.htm
                                             json.pixels[index] = grayscale[0];
          in_class_code.htm
          loops.html
          poperators.html
          pixels.json
                                         saveJSON(ison, 'crv.ison'):
```

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





```
p5*
                    Sketch ▼ Help & Feedback ▼
                                                                                 Hello, smmike! | My Account ▼
      File w
             Edit v
              Tidy Code
                            û+Tab
                                  vester brace / by smmike
              Find
                           Ctrl+F
              Find Next
                           Ctrl+G
              Find Previous 12+Ctr1+G
       let num = 0:
       function setup(){
         createCanvas(200,200);
         fill(128);
    8 function draw() {
          // Loop thru 0 - 256 for color
         let brightness = num % 256;
   10
         background(brightness);
   12
         // Loop thru 0 - 200 for position
         stroke(256 - brightness);
   14
         let xv = num % 200;
   16
         rect(xy, xy, 20, 20);
         num++; // increment by 1
   19
```

p5.js editor also offers convenient code formatting functions.



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



