Animation and Functions

Create a new temporary project (you don't need to save it). Type in this piece of code:

```
// declare global variable accessible from any part of the code.
// global variables are declared outside of any functions.
globalVarX = 50;

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

function draw() {
  background(220);
  background(255, 255, 255);
  fill(255, 0, 0);
  ellipse(globalVarX, 100, 30, 30);
  globalVarX = globalVarX + 1;
}
```

The `draw` function is automatically called 60 times per second!

Tasks

- Try changing 50 in `globalVarX` to different number.
- Try changing the line `x = x + 1` to `x = x - 1` or to `x = x + 5`
- Try moving the call to `background` from `draw` to `setup` - what happens?

NOTE!!:

When using `setup/draw` it is not allowed to call drawing functions outside of `setup` and `draw`. Everything must be moved into the two functions.

Aquarium continues

- *Rewriting the aquarium project to use `setup/draw`:*
  - Add empty `setup` and `draw` functions to the bottom of the program
  - Call `createCanvas(400, 400)` in `setup`
  - Call all the drawing functions in `draw`, incl. drawing of the background
- *Make the fish swim:*
  - Create two global variables `fish1X` and `fish2X` (before `setup/draw`)
  - Use the new variables as x-argument when you call `drawFish()`
  - Update the variables with $+1/-1$ inside the `draw` function

DIKU
ds@di.ku.dk

Green City continues

- Create two global variables `carX` and `cloudX`
- Make the car drive to the right
- Make the cloud start outside the image on the right side and move to the left

Randomness

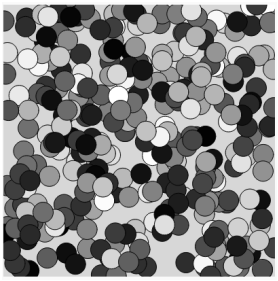
Create a brand new project and save it as "random_circles". Add the following code:

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

function draw() {
  // create 2 variables for different x
      y coordinates on the canvas
  x = random(0, width);
  y = random(0, height);

  // color the circle with a random
      monochrome shade
  fill(random(255));

  // draw the ellipse every frame
  ellipse(x, y, 30, 30);
}
```



Tasks

- Make the circles change size randomly
- Make the circles be drawn in random colours. Experiment until you find a colour scale that you think is nice.

DIKU
ds@di.ku.dk

Interaction with the Mouse

The position of the mouse can be read with the variables `mouseX` and `mouseY`. A kind of drawing program can be written like this:

```
function setup() {
  createCanvas(800, 800);
  background(255, 204, 0);
}

function draw() {
  fill(0, 0, 0);
  ellipse(mouseX, mouseY, 5, 5);
}
```

Keyboard input

Open the drawing program project and try adding the following new function:

```
function setup() {
  createCanvas(400, 400);
  background(0);
}

function draw() {

}

function keyPressed(){
  background(0,0,255,255);
}
```

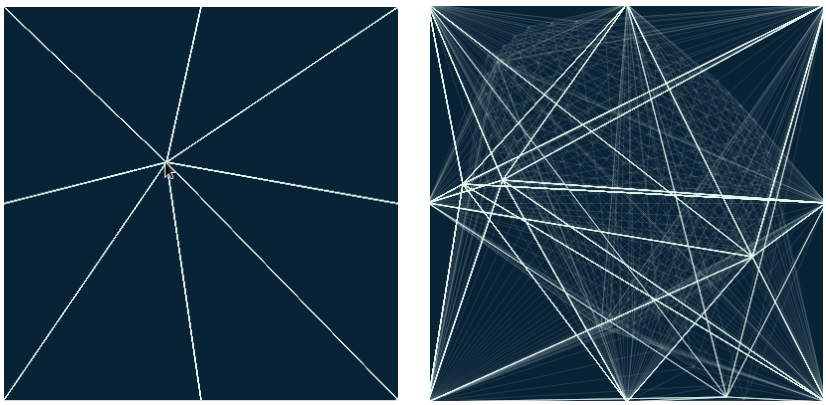
Start the program and press any key on the keyboard. To check for a specific key, the variable "`key`" can be read, like in the following code. It can be added to the `keyPressed()` function.

```
if (key == 'c') { // Check if the 'c' key is pressed
   background(255, 204, 0); // Set background color to a blue hue
```


DIKU
ds@di.ku.dk

New Project:Creative Programming

The task is now to create a more elaborate drawing program. Sign lines from all corners and midpoints on the sides. Use the variables `width` and `height`.



Remember to save the project!

DIKU
ds@di.ku.dk

Introduction: Conditionals

With conditions, we can make things happen when special criteria are met met. Open the aquarium project and try inserting the following into it The `draw` function:

```
if (fish1X > 500) {
    fish1X = -40;
}
if (fish2X < -50) {
fish2X = 200;
}
```

What happens?

Conditionals

In conditionals, we define a condition that is checked each time the code runs. This can be any condition (e.g. $Fish1X \geq 10$) that evaluates to either true or false. When the condition evaluates to true, the code in the brackets runs. If the condition is not met, an alternative outcome can be defined by an else statement. See the following syntax:

```
if (condition) {
  //Code executed if the condition is true
} else {
  //Code executed if the condition is false
}
```

Change direction

By making a variable that contains the direction in which the fish swims, we can change the direction when it reaches the sides.

- Define a global variable `fish1XVelocity` and set it to 1
- Change `fish1X = fish1X + 1` to `fish1X = fish1X + fish1XVelocity`
- Remove the previous `fish1X` condition
- Add these conditions:

```
if (fish1X > 400){
    fish1XVelocity = -1
}
if (fish1X < 0){
    fish1XVelocity = 1
}
```


DIKU
ds@di.ku.dk

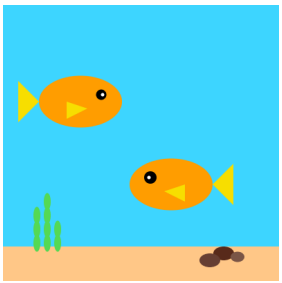
Change appearance

Try changing the `drawFish` function to strip the direction with the fish as an argument and use a condition to draw the fins and The eyes of the fish differ depending on the direction it is swimming.

```
function drawFish(x, y, eyeSize,
    velocity){
    ... draw body ...

    // Swim to the right
    if (velocity >= 0){
        ...draw fins and eyes ...
    }

    // Swim to the left
    if (velocity < 0){
        ...draw fins and eyes ...
    }
}
```



Finish the aquarium project

Now, we are almost done with the aquarium project. Add, if necessary, more elements. For example bubbles that emerge from the bottom and swim towards the surface, a chest with gold or whatever you want in your aquarium.

DIKU
ds@di.ku.dk

Animation example

You can use this code as a stepping stone to move forward

```
// ####### GLOBAL VARIABLES #######
var fish1X = 50;
var fish1Velocity = 1;
var fish2X = 500;
var fish2Velocity = -1;

// Canvas setup
function setup() {
  createCanvas(400, 400);
}

// ####### DRAWING FUNCTIONS #######

function drawFish(fishX, fishY, eyeSize, velocity) {
  // TODO: Complete this function to draw a fish
  // Remember to draw different fins/eyes based on velocity
      direction

  noStroke();
  fill(255, 157, 0);  // Orange body
  ellipse(fishX, fishY, 120, 75);

  if (velocity >= 0) {
    // Swimming right - draw fins and eye on right side
    // YOUR CODE HERE: Add tail, fins, and eye for right-facing fish

  } else {
    // Swimming left - draw fins and eye on left side
    // YOUR CODE HERE: Add tail, fins, and eye for left-facing fish

  }
}

function drawStones(stonesX) {
  // TODO: Draw a group of stones at the given x position
  // Hint: Use multiple ellipses with brown colors

  // YOUR CODE HERE
}
```

Animation example continued

```
function drawSeaweed(seaweedX) {
  // TODO: Draw seaweed plants at the given x position
  // Hint: Use green ellipses stacked vertically

  // YOUR CODE HERE
}

function drawScene() {
  // Draw the aquarium background
  background(61, 213, 255);  // Blue water

  // Draw sandy bottom
  fill(255, 199, 135);  // Sandy color
  noStroke();
  rect(0, 350, 400, 200);

  // Add decorations
  drawStones(300);
  drawSeaweed(65);
}

// ###### END DRAWING FUNCTIONS ######

// ####### CONTROL FUNCTIONS #######

function moveFish1() {
  // Update fish1 position
  fish1X = fish1X + fish1Velocity;

  // Check boundaries and reverse direction
  if (fish1X > 500) {
    fish1Velocity = -fish1Velocity;
  }
  if (fish1X < 0) {
    fish1Velocity = -fish1Velocity;
  }
}

function moveFish2() {
  // TODO: Complete this function for fish2
  // Should work similar to moveFish1 but use fish2 variables

  // YOUR CODE HERE
}

// ######## END CONTROL FUNCTIONS #######
```


DIKU
ds@di.ku.dk

Animation example continued

```
// ########## MAIN PROGRAM ##########

function draw() {
  // Update positions
  moveFish1();
  moveFish2();

  // Draw everything
  drawScene();
  drawFish(fish1X, 140, 15, fish1Velocity);
  drawFish(fish2X, 260, 30, fish2Velocity);
}
```

More exercises about conditions

Open the Green City project and do the following:

- Make the cloud move back to the start and pass by again and again, each time it hits the edge
- Make the car turn over at both ends.

Tasks

The car must stop when the battery is empty

- Define a global variable `carBattery` and set it to 100
- Decrease it by 0.1 each time `draw` is called
- Display the battery status using the `text(string, x, y)` function. Remember to convert the number to a string using `str()`.
- If the battery is empty (`<= 0`) the car must stop (set velocity to 0)

Add a `keyPressed()` function and make it possible to charge the electric car when you press 'C':

- Add `0.3` to `carBattery` every time 'C' is pressed on the keyboard
- If `carBattery` exceeds 100, set it to 100 so you can't charge more than 100%

DIKU
ds@di.ku.dk

Changing wind speed

The variable `frameCount` counts how many times `draw` is run since the program started. Enter this in the draw function i the electric car project:

```
fill(0, 0, 0);
text(frameCount, 350, 20);

if (frameCount % 60 == 0){
    print(frameCount)
}
```

Since `draw` is executed at *60 frames per second*, the **`print`** function will be executed every second.

We can also use that in the electric car project to update the wind speed every second.

- Create a global windSpeed variable
- Update windSpeed with a new random value every second
- Display the wind speed using `text()`
- Create a global variable `powerplantOn`
- Make it possible to turn the power plant on and off with a press of 'p'
- Draw clouds of smoke over the power plant when it is on
- Create a variable `co2emission` and add a bit as long the power plant is on. Display the value with `text()`

DIKU
ds@di.ku.dk

Sounds

In this small worksheet, we will add sound files to make sound effects!

1. First check reference https://p5js.org/reference/p5.MediaElement/play/
2. Then download the sound files from ABASALON
3. Also check out the sound sketch also uploaded
4. Create a new file named `sound01`

```
function preload() {
    // Load the sound files before the program starts
    // Load the first sound file (e.g., a music track or sound
       effect)
    sound = loadSound('sound.mp3');
    // Load the second sound file (e.g., a ball tap sound)
    sound1 = loadSound('balltap.wav');
}

function setup() {
  // Setup the canvas size
  createCanvas(400, 400);  // Create a canvas of 400x400 pixels
}

function draw() {
  // Draw the background and instructions
  background(220);  // Set the background color to light grey (RGB
     value: 220)
  // Align text to the center horizontally and vertically
  textAlign(CENTER, CENTER);
  textSize(24);  // Set the text size to 24 pixels
  // Display the instructions in the center of the canvas
  text('Press "S" or "A" to play sound', width / 2, height / 2);
}

function keyPressed() {
  // Detect key presses and play the corresponding sound
  if (key === 's' || key === 'S') {
    // Play the first sound when the 'S' key is pressed (case
       insensitive)
    sound.play();
  }
  if (key === 'a' || key === 'A') {
    // Play the second sound when the 'A' key is pressed (case
       insensitive)
    sound1.play();
  }
}
```

Descriptions

- **'preload()' function**: Loads the sound files into memory before the sketch starts. This ensures that the sounds are ready to be played when triggered.
- **'setup()' function**: Sets up the canvas where the sketch will be displayed, specifying its size (400x400 pixels).
- **'draw()' function**: Continuously runs in a loop, updating the background color and displaying the instruction text centered on the canvas.