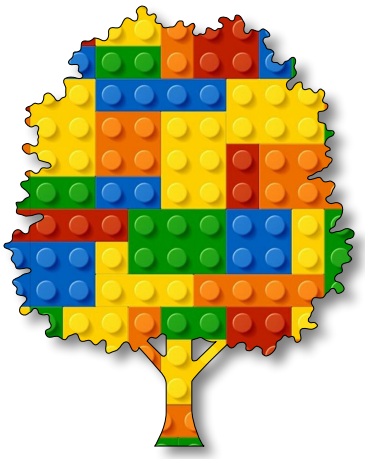
Algorithmic Art

Module F

Unit #7

telling the time



Content

Module F Unit #7 telling the time

Introduction

A brief look at how you can incorporate the `time` and `date`, and using the `millis()` function to measure the passage of time.

This unit is about getting the time and date from your computer. What it also serves is to look at some other features like text alignment. One challenge that you could try is to create a seven-segment display using the time and a lot of code.

Sketch F7.1 every second counts

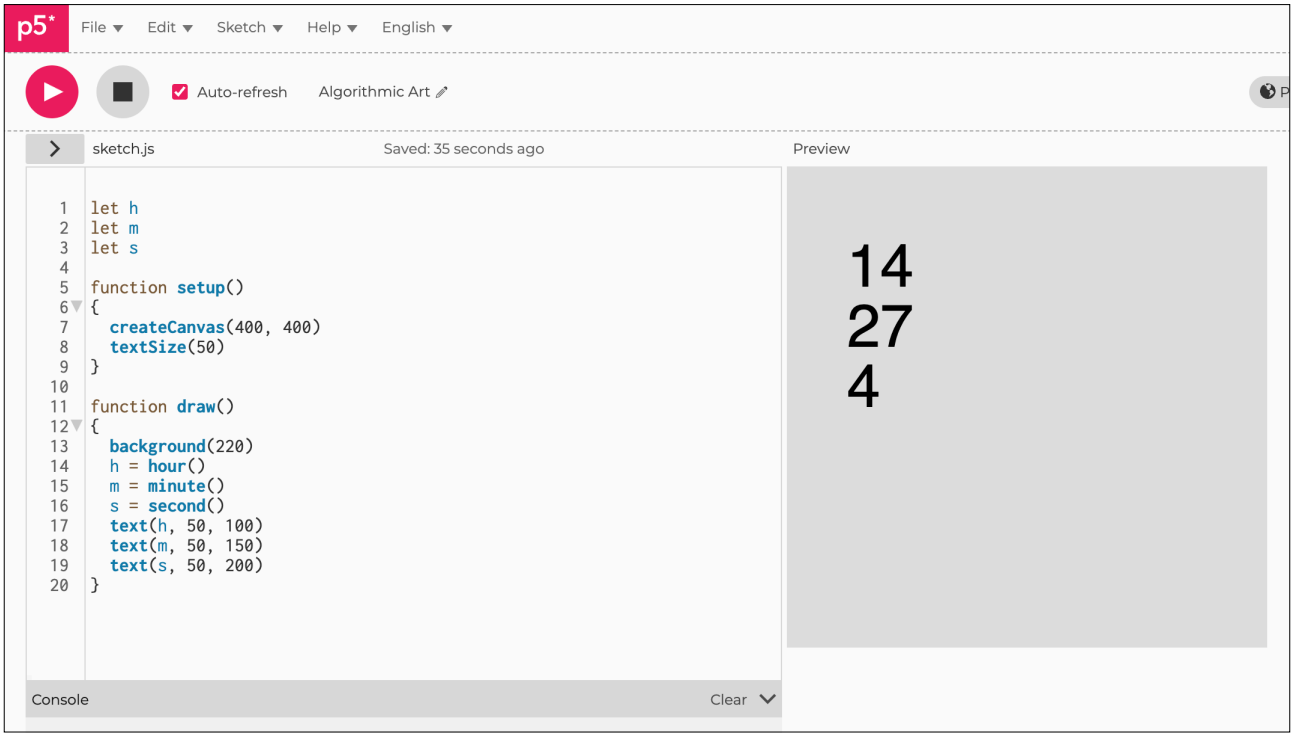
Getting the hour, minute, and seconds of the current time.

```
let h
let m
let s

function setup()
{
  createCanvas(400, 400)
  textSize(50)
}

function draw()
{
  background(220)
  h = hour()
  m = minute()
  s = second()
  text(h, 50, 100)
  text(m, 50, 150)
  text(s, 50, 200)
}
```

Notes

Gives the hour (24-hour clock), the minutes, and the seconds. Notice that it gives the minutes and seconds in single figures for below 10.

Figure F7.1


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 35 seconds ago
Preview
let h
let m
let s
function setup()
{
createCanvas(400, 400)
textSize(50)
}
function draw()
{
background(220)
h = hour()
m = minute()
s = second()
text(h, 50, 100)
text(m, 50, 150)
text(s, 50, 200)
}
14
27
4
Console
Clear

Time functions

`hour()` gives you the current hour.
`minute()` gives you the current minutes.
`second()` gives you the current seconds.

🗒️ Notes

Collects data from the internet.

🌻 Challenge

How would you put text in front (hint: `text('hours: ' + h, 50, 100)`

Sketch F7.2 a second look

❗ Replacing `text(m)` and `text(s)`.
A better-looking clock.

```
let h
let m
let s

function setup()
{
  createCanvas(400, 400)
  textSize(50)
}

function draw()
{
  background(220)
  h = hour()
  m = minute()
  s = second()
  text(h, 50, 100)
  if (m < 10)
  {
    text('0' + m, 50, 150)
  }
  else
  {
    text(m, 50, 150)
  }
  if (s < 10)
  {
    text('0' + s, 50, 200)
  }
  else
  {
    text(s, 50, 200)
  }
```

```
}
```

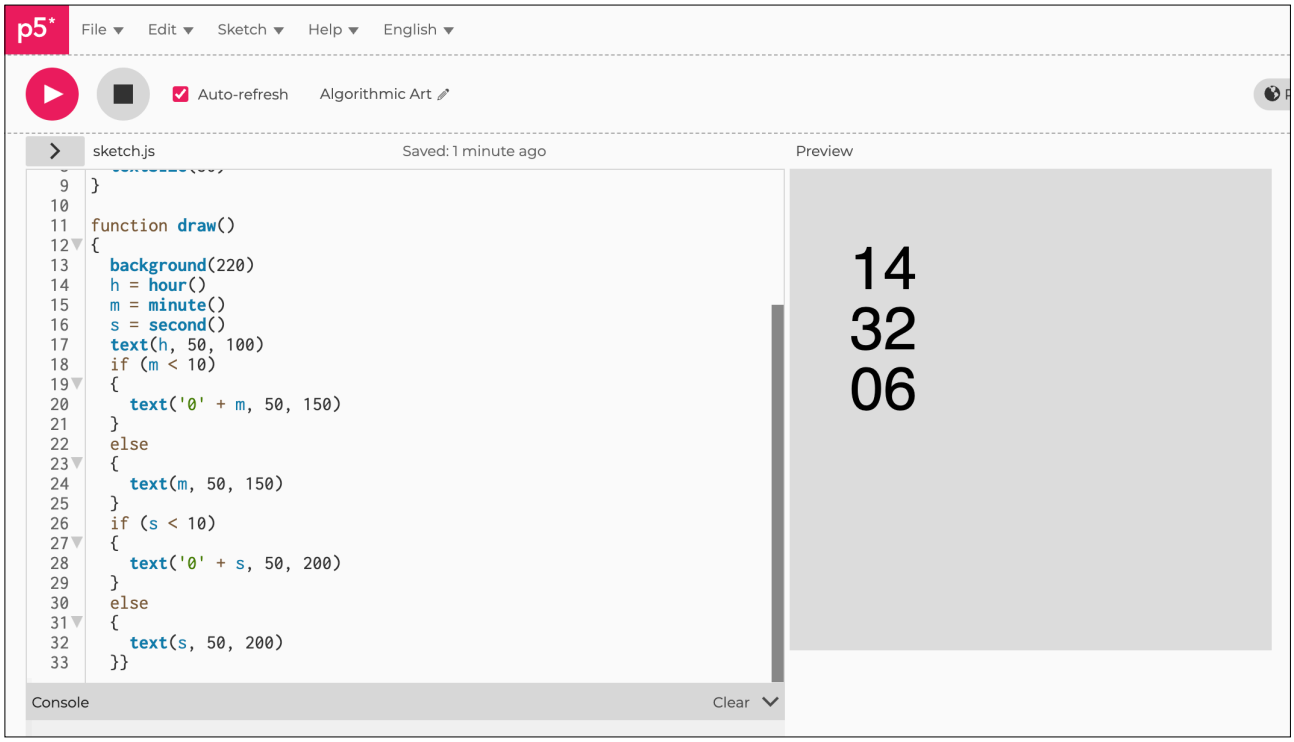
🗒️ Notes

Improves the look by putting the 0 in front of the digit if less than 10 for minutes and seconds.

🌻 Challenge

1. Now put 0 before the hour (if earlier than 10 o'clock).
2. What if you don't want the 24-hour clock?

Figure F7.2



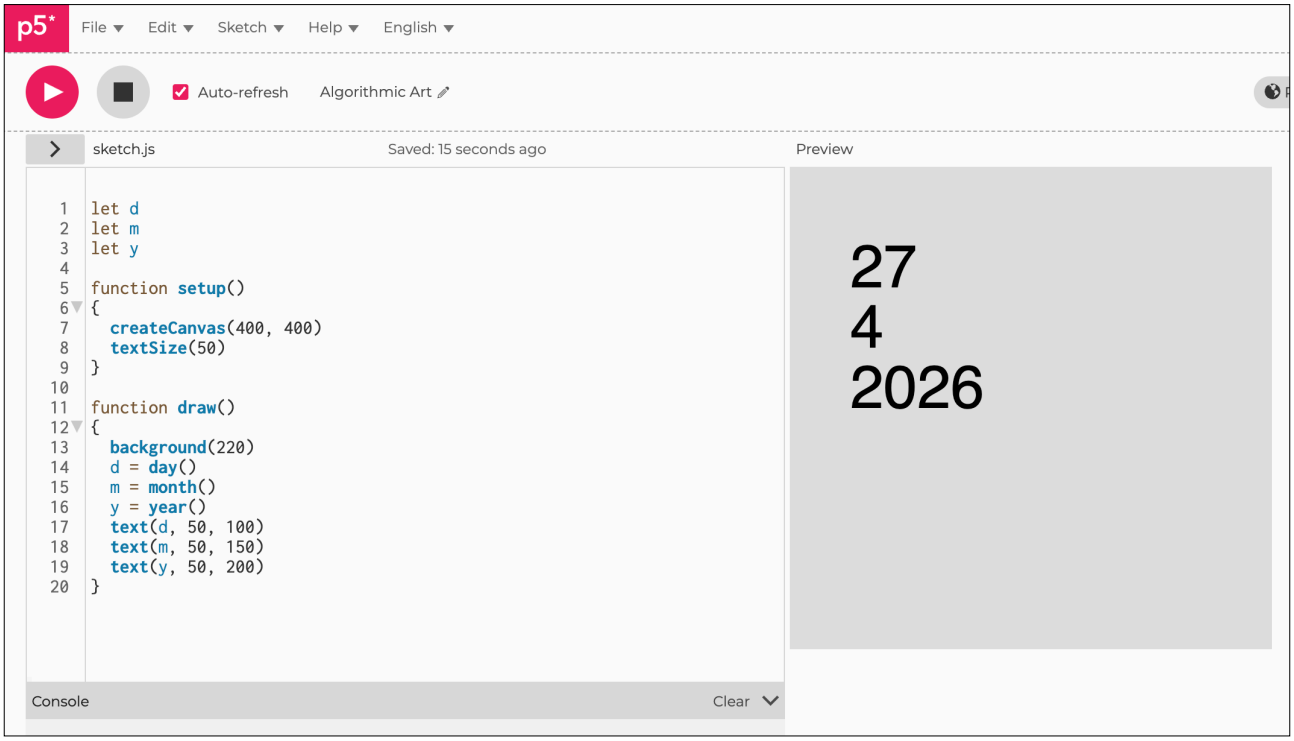
Sketch F7.3 what a year

❗ Start a newish sketch.
Adding the date.

```
let d
let m
let y

function setup()
{
  createCanvas(400, 400)
  textSize(50)
}

function draw()
{
  background(220)
  d = day()
  m = month()
  y = year()
  text(d, 50, 100)
  text(m, 50, 150)
  text(y, 50, 200)
}
```

🗒️ Notes
Gives the day in numbers. The month in numbers, and also the year.

Figure F7.3


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let d
let m
let y
function setup()
{
createCanvas(400, 400)
textSize(50)
}
function draw()
{
background(220)
d = day()
m = month()
y = year()
text(d, 50, 100)
text(m, 50, 150)
text(y, 50, 200)
}
27
4
2026
Console
Clear

Date functions

`day()` gives you the current day (as a number).
`month()` gives you the current month (as a number).
`year()` gives you the current year.

🗒️ Notes

These pull the data from the internet.

🌻 Challenge

Using the date and time, how creative could you be in showing all this information graphically?

Sketch F7.4 milliseconds of time

❗ Starting a new sketch.
Milliseconds of time represented differently, as a float, an integer, and as seconds.

```
function setup()
{
  createCanvas(400, 400)
  textSize(30)
}

function draw()
{
  background(220)
  text(millis(), 50, 50)
  text(int(millis()), 50, 100)
  text(int(millis()/1000), 50, 150)
}
```

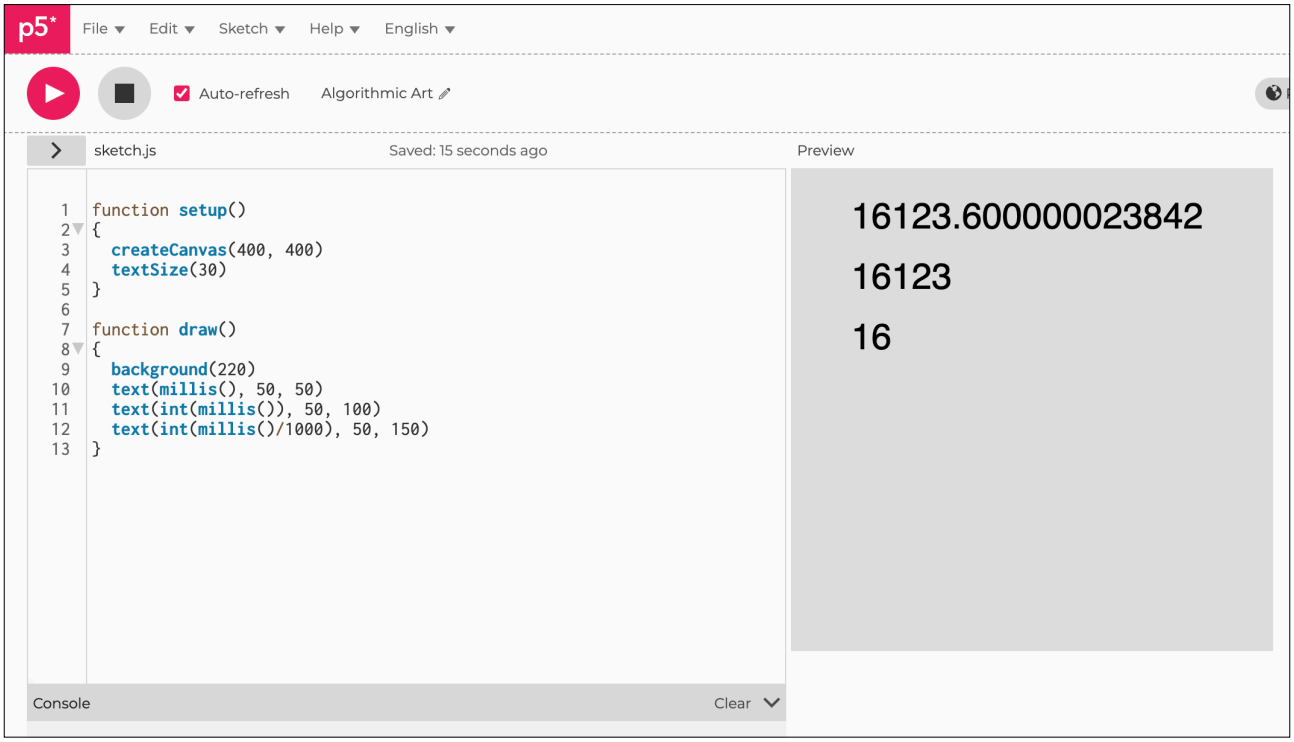
🗒️ Notes

The digits will appear as a blur. The millis() function returns the number of milliseconds that have elapsed since the programme started running as a `float`. It uses `int` to stop the numbers from having too many decimal points. The last one counts the number of seconds.

🌻 Challenge

Create a stopwatch effect where the draw() loop (using `noLoop()`) stops when you click the mouse.

Figure F7.4


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
function setup()
{
createCanvas(400, 400)
textSize(30)
}
function draw()
{
background(220)
text(millis(), 50, 50)
text(int(millis()), 50, 100)
text(int(millis()/1000), 50, 150)
}
16123.600000023842
16123
16
Console
Clear

Sketch F7.5 in the blink of an eye

❗ Start a new sketch.

This is a rather long-winded way of a blinking (every second) sketch, but it demonstrates how it can be done. You could do it for any length of time.

```
let a = 0
let b = 0
let c = 0
let value = 255

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

function draw()
{
  b = millis()
  c = b - a
  fill(value)
  circle(width/2, height/2, 100)
  if (c <= 1000)
  {
    value = 255
  }
  if (c >= 1000)
  {
    value = 0
  }
  if (c >= 2000)
  {
    a = b
  }
}
```

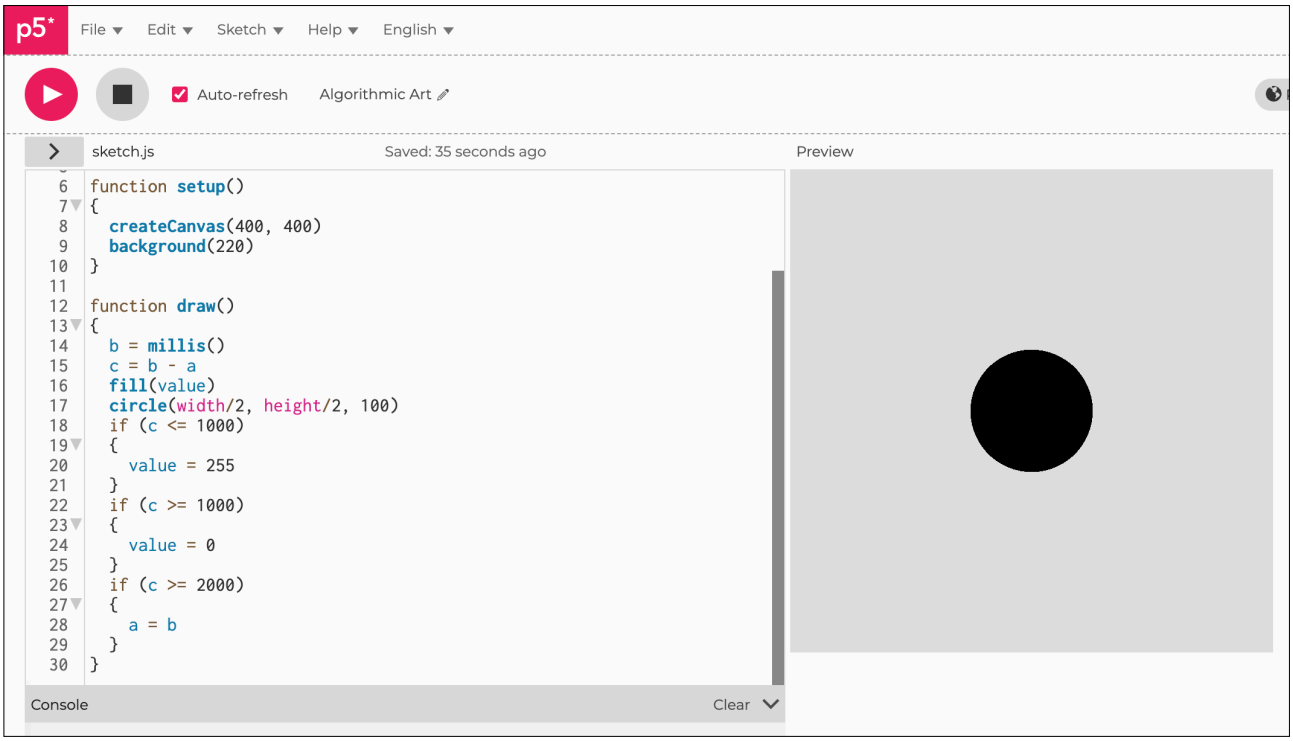
🗒️ Notes

A simple blink programme. Uses a bit of simple maths to count 1,000 milliseconds (1 second).

🌻 Challenges

1. Make it blink faster or slower.
2. Create a slider to change the rate of blinking.

Figure F7.5



Sketch F7.6 alternative modulo

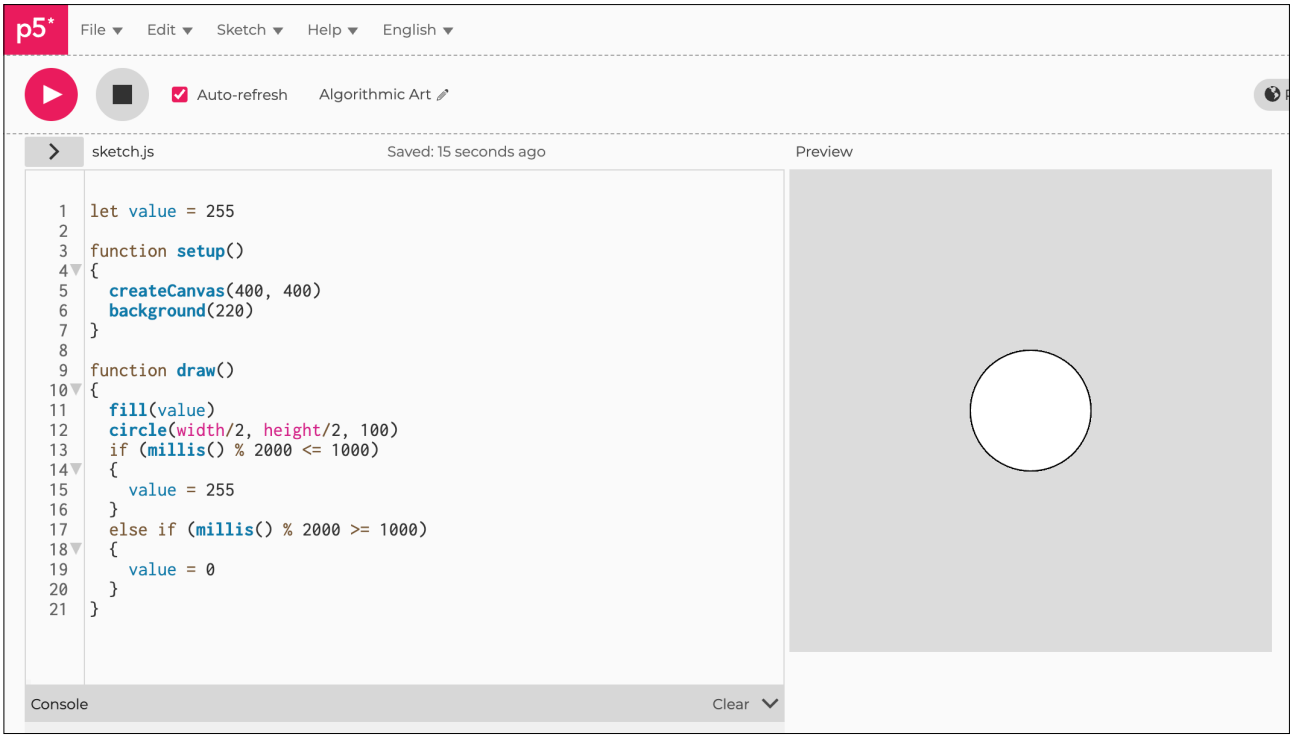
❗ Start a new sketch.
This does the same thing where `modulo` returns the remainder.

```
let value = 255

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

function draw()
{
  fill(value)
  circle(width/2, height/2, 100)
  if (millis() % 2000 <= 1000)
  {
    value = 255
  }
  else if (millis() % 2000 >= 1000)
  {
    value = 0
  }
}
```

🗒️ Notes

The same result: the circle blinks black and white every second but a much simpler code.

Figure F7.6


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let value = 255
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
fill(value)
circle(width/2, height/2, 100)
if (millis() % 2000 <= 1000)
{
value = 255
}
else if (millis() % 2000 >= 1000)
{
value = 0
}
}
Console
Clear

Sketch F7.7 making a clock

We will do this in stages so you get an idea of how the bits go together. First off, let's draw the clock.

```
let radius = 200
let clockDiameter

function setup()
{
  createCanvas(400, 400)
  clockDiameter = radius * 1.7
  stroke('teal')
  strokeWeight(2)
}

function draw()
{
  background('orange')
  translate(width/2, height/2)
  fill('lightyellow')
  circle(0, 0, clockDiameter)
}
```

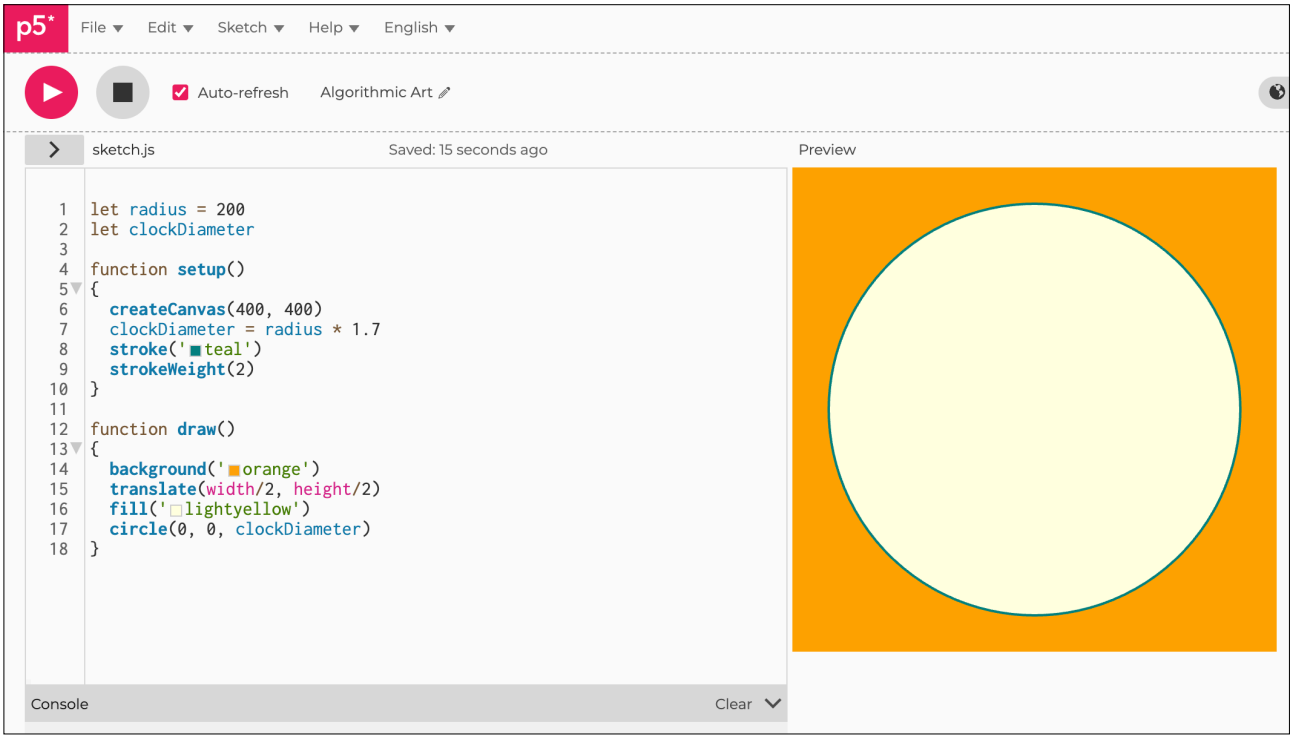
🗒️ Notes

Simple circle on a simple background. The radius is our reference point for all other diameters/radii.

🌻 Challenge

Feel free to use other colours.

Figure F7.7


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let radius = 200
let clockDiameter
function setup()
{
createCanvas(400, 400)
clockDiameter = radius * 1.7
stroke('teal')
strokeWeight(2)
}
function draw()
{
background('orange')
translate(width/2, height/2)
fill('lightyellow')
circle(0, 0, clockDiameter)
}
Console
Clear

Sketch F7.8 other radii

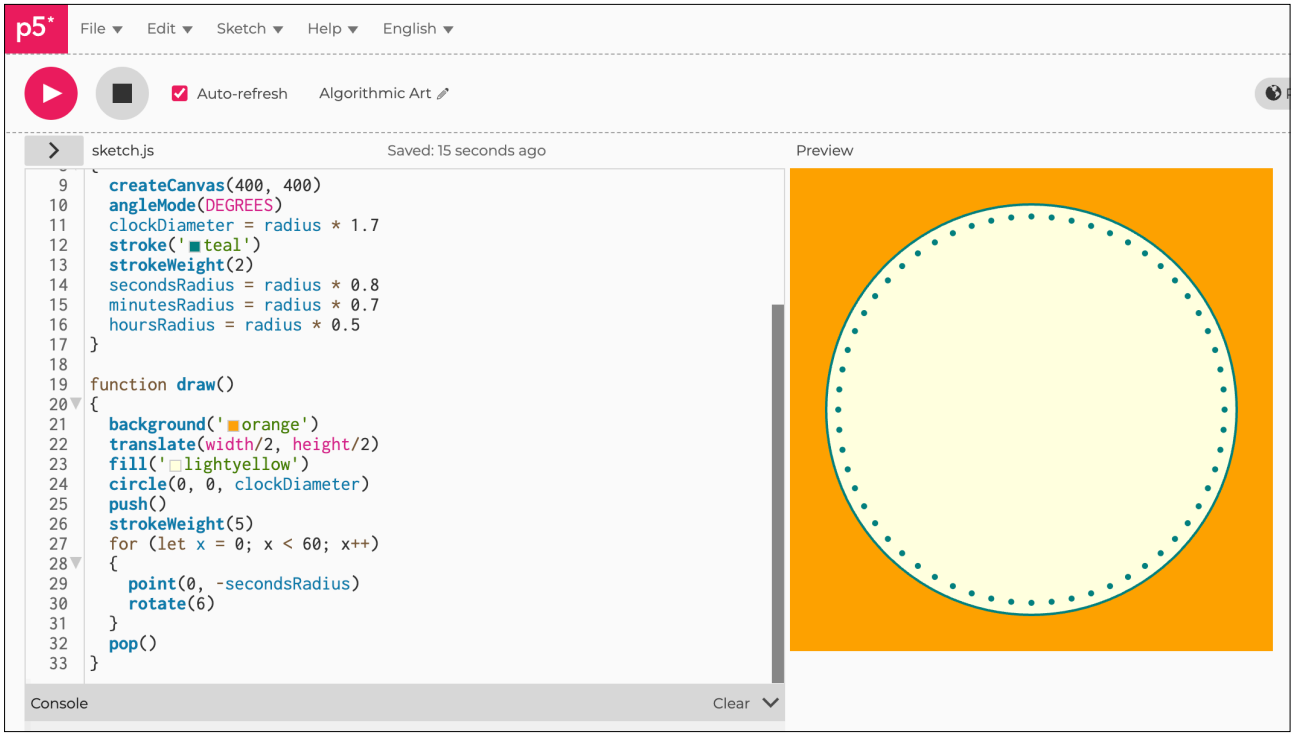
Set the radii for the seconds, minutes, and hours. We draw the ticks for the seconds; we are going to work in degrees.

```
let radius = 200
let clockDiameter
let secondsRadius
let minutesRadius
let hoursRadius

function setup()
{
  createCanvas(400, 400)
  angleMode(DEGREES)
  clockDiameter = radius * 1.7
  stroke('teal')
  strokeWeight(2)
  secondsRadius = radius * 0.8
  minutesRadius = radius * 0.7
  hoursRadius = radius * 0.5
}

function draw()
{
  background('orange')
  translate(width/2, height/2)
  fill('lightyellow')
  circle(0, 0, clockDiameter)
  push()
  strokeWeight(5)
  for (let x = 0; x < 60; x++)
  {
    point(0, -secondsRadius)
    rotate(6)
  }
  pop()
}
```

Notes

We have our foundation; next, we need to draw the hands. We use `push()` and `pop()` so that we do not compound the rotation each time.

Figure F7.8



Sketch F7.9 the angles

Calculating the angle from the time.

```
let radius = 200
let clockDiameter
let secondsRadius
let minutesRadius
let hoursRadius
let secondsAngle
let minutesAngle
let hoursAngle

function setup()
{
  createCanvas(400, 400)
  angleMode(DEGREES)
  clockDiameter = radius * 1.7
  stroke('teal')
  strokeWeight(2)
  secondsRadius = radius * 0.8
  minutesRadius = radius * 0.7
  hoursRadius = radius * 0.5
}

function draw()
{
  background('orange')
  translate(width/2, height/2)
  fill('lightyellow')
  circle(0, 0, clockDiameter)
  push()
  strokeWeight(5)
  for (let x = 0; x < 60; x++)
  {
    point(0, -secondsRadius)
    rotate(6)
  }
```

```
  pop()
  secondsAngle = map(second(), 0, 60, 0, 360)
  minutesAngle = map(minute(), 0, 60, 0, 360)
  hoursAngle = map(hour(), 0, 12, 0, 360)
}
```

🗒️ Notes

Nothing new to see here; we are just calculating the angles by mapping the number of seconds, minutes, and hours to 360.

Sketch F7.10 drawing the hands

Now we will draw the seconds hand, the minutes hand, and the hours hand.

```
let radius = 200
let clockDiameter
let secondsRadius
let minutesRadius
let hoursRadius
let secondsAngle
let minutesAngle
let hoursAngle

function setup()
{
  createCanvas(400, 400)
  angleMode(DEGREES)
  clockDiameter = radius * 1.7
  stroke('teal')
  strokeWeight(2)
  secondsRadius = radius * 0.8
  minutesRadius = radius * 0.7
  hoursRadius = radius * 0.5
}

function draw()
{
  background('orange')
  translate(width/2, height/2)
  fill('lightyellow')
  circle(0, 0, clockDiameter)
  push()
  strokeWeight(5)
  for (let x = 0; x < 60; x++)
  {
    point(0, -secondsRadius)
    rotate(6)
```

```
  }
  pop()
  secondsAngle = map(second(), 0, 60, 0, 360)
  minutesAngle = map(minute(), 0, 60, 0, 360)
  hoursAngle = map(hour(), 0, 12, 0, 360)

  push()
  rotate(secondsAngle)
  stroke('darkred')
  line(0, 20, 0, -secondsRadius)
  pop()

  push()
  strokeWeight(3)
  rotate(minutesAngle)
  stroke('blue')
  line(0, 0, 0, -minutesRadius)
  pop()

  push()
  strokeWeight(5)
  rotate(hoursAngle)
  stroke('grey')
  line(0, 0, 0, -hoursRadius)
  pop()

  stroke('darkred')
  circle(0, 0, 5)
}
```

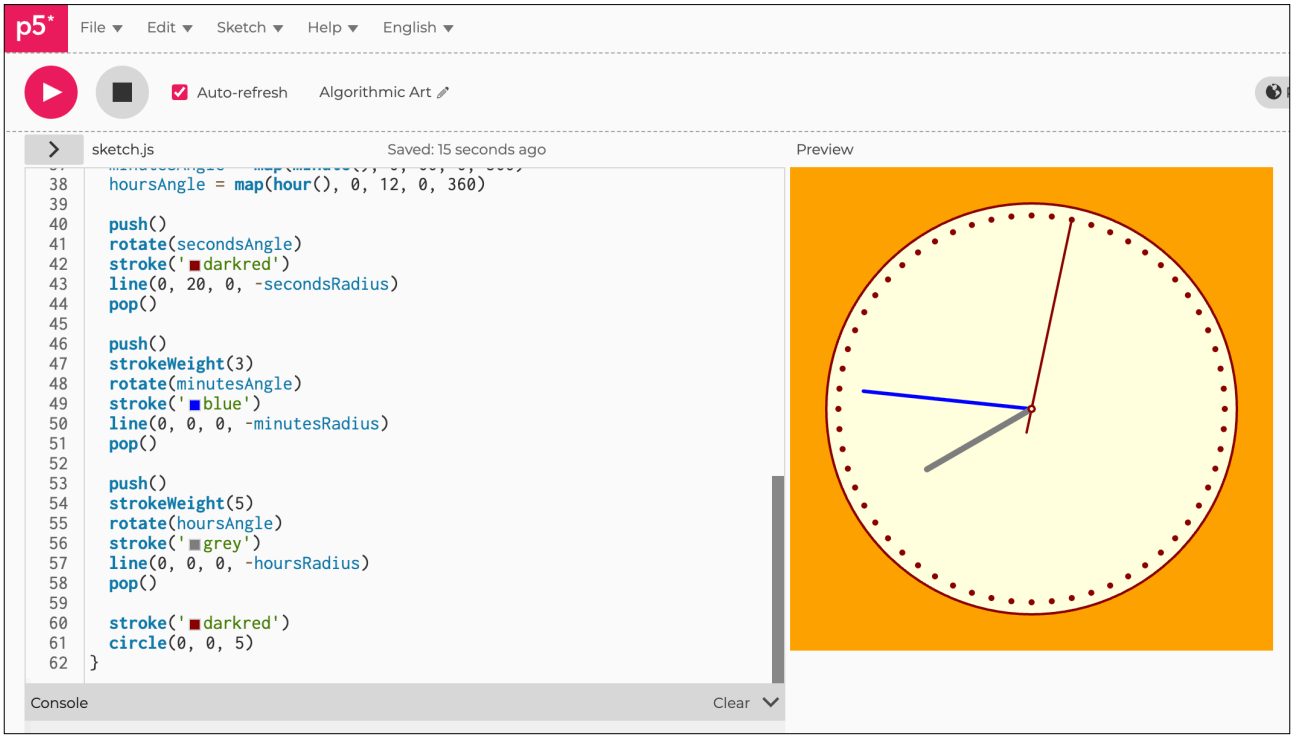
🗒️ Notes

We have our finished clock.

🌻 Challenge

Can you think of some creative ways to illustrate the time, for instance, using lines or 3D shapes?

Figure F7.10


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
38 hoursAngle = map(hour(), 0, 12, 0, 360)
39
40 push()
41 rotate(secondsAngle)
42 stroke('darkred')
43 line(0, 20, 0, -secondsRadius)
44 pop()
45
46 push()
47 strokeWeight(3)
48 rotate(minutesAngle)
49 stroke('blue')
50 line(0, 0, 0, -minutesRadius)
51 pop()
52
53 push()
54 strokeWeight(5)
55 rotate(hoursAngle)
56 stroke('grey')
57 line(0, 0, 0, -hoursRadius)
58 pop()
59
60 stroke('darkred')
61 circle(0, 0, 5)
62 }
Console
Clear