p5.js English Cheat Sheet

Lifecycle Functions

preload() {}: The external resources are defined in body of this function. The other functions can't be started until all resources are loaded.

```
function preload() {
    model = loadModel('teapot.obj');
    img = loadImage('cat.jpg');
}
```

setup() {}: The first definitions are located in body of this function.

```
function setup() {
    createCanvas(640, 480);
}
```

draw() {}: This function contains actions and drawings of the game loop in its body.

```
function draw() {
    x++;
    rect(x, 100, 50, 50);
}
```

Drawing Functions

createCanvas(): Creates the drawing area.

```
createCanvas(width, height) // For 2D drawing
createCanvas(width, height, WEBGL) // For WEBGL mode
```

line(): Draws a line.

```
line(x1, y1, x2, y2)
```

rect(): Draws a rectangle.

```
rect(x, y, width, height)
```

circle(): Draws a circle.

```
circle(x, y, diameter)
```

ellipse(): Draws an ellipse.

```
ellipse(x, y, diameter)
ellipse(x, y, width, height)
```

text(): Shows a text.

```
text('Content', x, y[, horizontalTextBoundary, verticalTextBoundary])
```

rectMode(): Sets the locating options of the rectangles which will draw.

```
rectMode(CORNER) // Sets the upper left corner as the beginning point and adds the length values to beginning coordinates

rectMode(CORNERS) // Sets the upper left corner as the beginning point and the length parameters are selected as end of the drawing

rectMode(CENTER) // The selected location parameters are be the center point of the rectangle and the size of the rectangle comes from the length parameters

rectMode(RADIUS) // The selected location parameters are be the center point of the rectangle and the length parameters sets the half of the size values of the rectangle
```

ellipseMode(): Works the same as `rectMode()` for ellipses and circles.

```
rectMode(CORNER | CORNERS | CENTER | RADIUS)
```

fill(): Sets the fill color of the shape. It's used before the drawing shape functions.

```
fill(255); // Grayscale tones
fill(255, 255, 255); // RGB
fill([255, 255, 255]); // RGB array
fill(255, 255, 255, 255); // RGBA
fill([255, 255, 255, 255]); // RGBA array
fill('white'); // HTML color code name
```

stroke(): Sets the border color of the shape. It's used same as the `fill()`.

```
stroke(255, 255, 255); // RGB
```

noFill(): Makes the shapes has no fill color.

noStroke(): Makes the shapes has no borders.

frameRate(): Sets how many frames are drawn in a second.

```
frameRate(25);
```

General Functions

random(): Generates a random float number between two values.

```
randomNumber = random(2, 5)
```

int(): Turns the parameter it gets into an integer number.

```
integerNumber = int(2.6)
```

float(): Turns the parameter it gets into a float number.

```
floatNumber = float(4)
```

str(): Turns the parameter it gets into a string.

```
content = str(8)
```

abs(): Returns the absolute value.

```
number = abs(-3)
```

sin(): Sine function.

cos(): Cosine function.

tan(): Tangent function.

degrees(): Turns the radian value into degrees.

radians(): Turns the degrees value into radians.

angleMode(): Sets the use of angle parameter for the functions which have angle parameter.

```
angleMode(DEGREES | RADIANS)
```

createVector(): Creates a vector.

```
angleMode(x, y[, z])
```

constrain(): Constrains the value of a variable between two values.

```
newValue = contrain(variable, lowerBoundary, upperBoundary)
```

map(): Adjusts the value of a variable that can take a value between two boundaries to two new boundary values.

```
newValue = map(value, currentLowerBoundary, currentUpperBoundary, newLowerBoundary,
newUpperBoundary)
```

Variables

frameCount: The number of the frames since the drawing started.

width: The width of the drawing area.

height: The height of the drawing area.

windowWidth: Page width.

windowHeight: Page height.

displayWidth: Screen width.

displayHeight: Screen height.

mouseX: The horizontal coordinate of the mouse.

mouseY: The vertical coordinate of the mouse.

pmouseX: The horizontal coordinate of the mouse from the previous frame.

pmouseY: The vertical coordinate of the mouse from the previous frame.

key: Value of the last key which pressed on the keyboard.