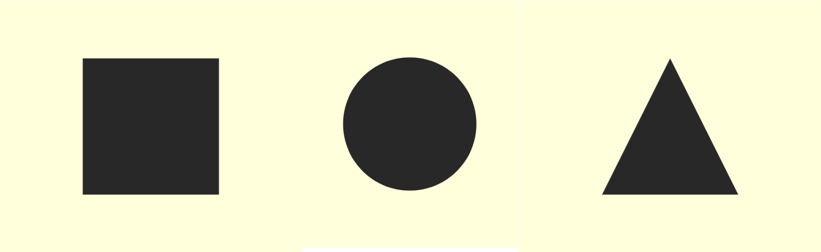
Basic shapes

As long as we have had the ability to draw, humans have used shapes for visual communication. Although shapes are not words, and therefore have no objective semantic meaning, we have a natural understanding of how to translate the characteristics of shapes into meaning: Cave paintings created more than 30,000 years ago can be appreciated today without a need for translation.

In this chapter, we will look at the three basic shapes: the rectangle, the ellipse, and the triangle. First, we will analyze the characteristics of each shape, and then demonstrate how to use these basic shapes in the design process.

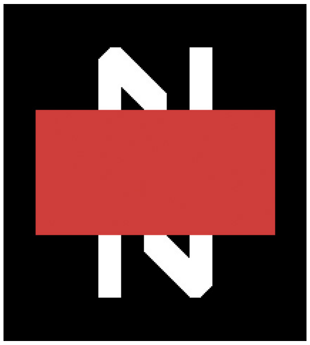


Rectangle

The rectangle is a symmetric, solid shape with parallel lines. As it does not exist much in nature, it has become the symbol for civilization itself. We build cities in rectangular grids, houses with bricks, and our interiors are rectangles too: doors, shelves, and windows.

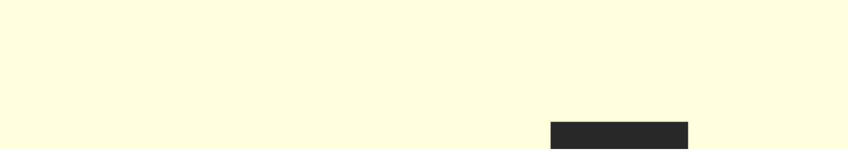
The rectangle has been used throughout the history of the arts to set up constraints for the artist. We use rectangular canvasses, and grid systems to further divide this canvas into smaller modules. In this digital age, we use rectangular screens and operate with a square as the smallest visual denomination: the pixel.

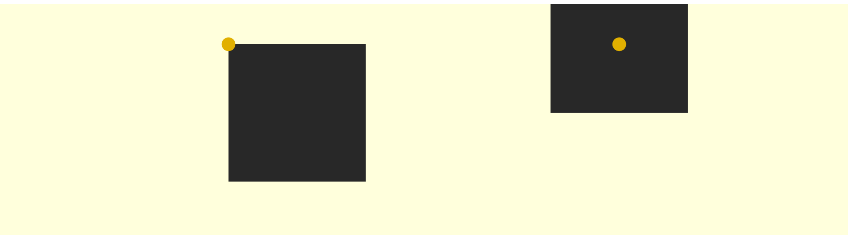
The squares in the Microsoft logo become a symbol for a window, a flag, and a pixelated screen. ©



The logo for the 8-bit musician Nullsleep has a rectangle on top of a pixelated font as a reference to both anti-authoritarianism and early computer art. ©

In geometry, a rectangle consists of four points connected to form a closed shape with internal angles of 90 degrees. However, the `rect()` function in P5 allows us to draw a rectangle simply by stating the position of the top left corner (which we will call the origin point •) as well as the size of the rectangle. As demonstrated below, the `rectMode()` function can be used to change the origin point of the rectangle to the center of the shape. This can be helpful in certain situations, e.g. if you want to draw a rectangle in the center of the canvas without needing to subtract half the size of the rectangle from its position.



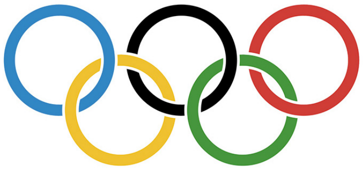


```
var size = width * 0.3;
rect(width/2, height/2, size, size);
```

```
rectMode(CENTER);
var size = width * 0.3;
rect(width/2, height/2, size, size);
```

Ellipse

The ellipse is a smooth shape found many places in nature, in the shape of planets, raindrops, and the eyes of most animals. With no apparent sense of direction, there is something neutral about the ellipse, and humans tend to gather in ellipses to achieve unity: We dance in circles, and design the seating of most parliaments in elliptical arrangements.

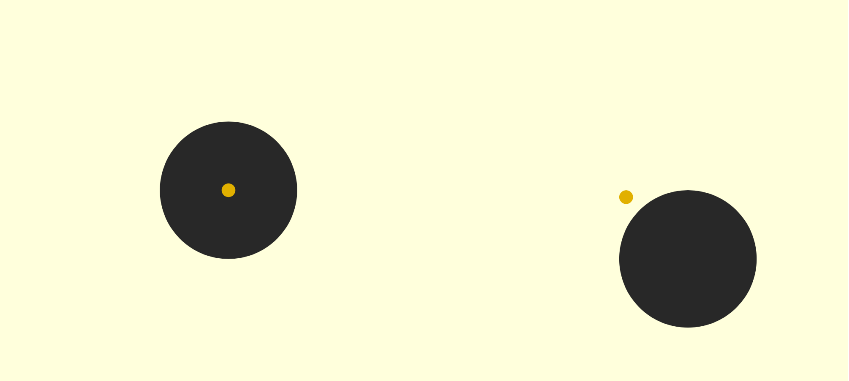


The symbol for the Olympic Games by Pierre de Coubertin consists of five connected ellipses with colors taken from the flags of the participating countries from the 1912 games. ©

The ABC logo by Paul Rand features a simple ellipse with an elliptical typeface. ©

In geometry, an ellipse is a closed shape where the following is true: Pick two locations inside the ellipse, and draw lines to any place on the outline.

No matter which location you pick on the outline, the combined length of your two lines will always be the same. Although the outline of an ellipse looks smooth to the human eye, computers actually draw ellipses as a series of short, straight, connected lines. Unlike `rect()`, the `ellipse()` function will draw an ellipse with an origin point in the center of the shape. As demonstrated below, the `ellipseMode()` function can be used to change the origin point to the top left corner. Given the nature of the ellipse, this means that the origin point is located outside the outline of the shape.



```
var size = width * 0.3;
ellipse(width/2, height/2, size, size);
```

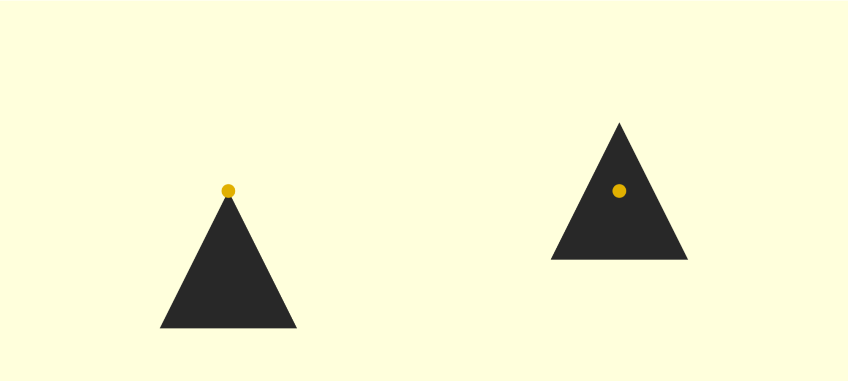
```
ellipseMode(CORNER);
var size = width * 0.3;
ellipse(width/2, height/2, size, size);
```

Triangle

The triangle is an asymmetric shape, unique among the basic shapes for its directionality. It is commonly known as the symbol for both masculinity (▲) and femininity (▼), and it is widely used in graphic design for its aesthetic qualities. One of our most significant cultural artifacts, the Great Pyramids of Giza, are also famous depictions of the triangle, pointing towards the assumed rotational center of the sky to which the egyptians ascribed godly qualities.

The Delta logo is a triangle consisting of four smaller triangles. The logo refers to the greek triangle letter by the same name, and the directionality of the triangle is used to imply speed and flight. ©

In geometry, a triangle is a closed shape consisting of three points. The sum of the internal angles of the triangle will always be 180 degrees (or 2PI radians). The triangle has played a central role in many mathematical breakthroughs, including Euclidian geometry, trigonometry, as well as 3D computer graphics. Unlike the `rect()` and `ellipse()` functions that both expect a single position for the shape's origin point, the `triangle()` function needs the coordinates of all three corners of the triangle. This also means that there is no such thing as a `triangleMode()` function. You will need to perform your own calculations to draw a triangle around a specific origin point, which is demonstrated in the examples below.



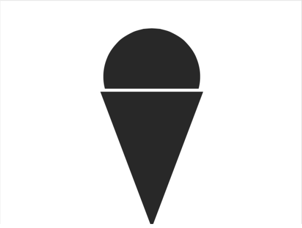
```
var size = width * 0.15;
translate(width/2, height/2);
triangle(0, 0, size, size*2, -size,
size*2);
```

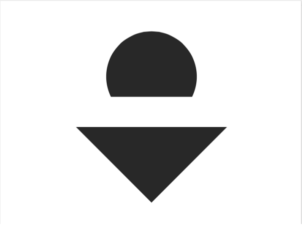
```
var size = width * 0.15;
translate(width/2, height/2);
triangle(0, -size, size, size, -size,
size);
```

An ice cream cone

I give my students the following (somewhat silly) exercise: Design an ice cream cone in black and white with only a single occurrence of each of the basic shape functions in the code. These tights constraints force the students to focus on the characteristics of the shapes, and how they can position, size, and rotate these shapes to achieve an effective design.

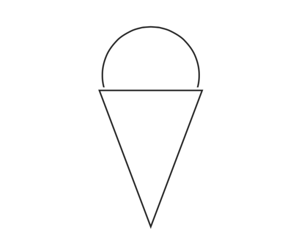
The most important aspect of this exercise is obviously to create a design that a majority of users will recognize as an ice cream cone. Whether or not a design accomplishes this is a rather objective task. Of the two designs below, it is clear that the first design manages to solve the assignment while the latter does not. It's also easy to analyze why: Although the shapes are almost identical between the two designs, the latter does not establish the proper visual relationships.



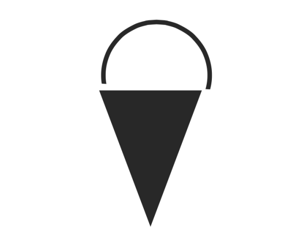


A second (and more subjective) aspect relates to the style of the design. To a certain degree, different styles fit different scenarios. If you're asked to create an icon for a website, an abstract style might be prefered, as simple designs work better in small sizes. On the other hand, if you're making an illustration for a children's book, something more bold and playful might be a better fit. Style can be used to serve a specific function, or it can be used purely for aesthetic qualities (which can also be a function, after all). Finally, the style of a design is where the subjective preferences of the designer is apparent.

In this exercise, I often encourage my students to practice designing in different styles, as it further develops their visual language. An important ingredient in the creation of style is the use of the `fill ()`, `stroke()` and `strokeWeight()` functions. As demonstrated below, these function can drastically alter the style of a design.

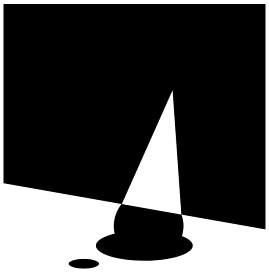


Equally proportioned shaped with the same stroke results in an abstract design.

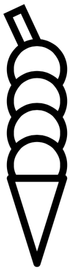


Thicker strokes combined with fills and rotation makes for a more playful design.

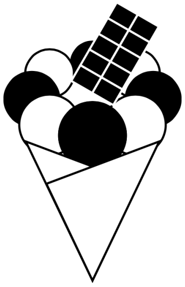
This exercise also encourages students to think systematically when implementing their designs in code. The three basic shape functions can only appear once in the code, so students need to use loops to draw more intricate designs. The following three examples are all by former students of mine, and they are displayed here as successful examples of all the things mentioned above: They all objectively solve the assignment of constructing an ice cream cone out of basic shapes. They all achieve widely different styles by using a clever combination of the basic shape functions and the following relationships: position, size, rotation, fill, line and stroke weight. Finally, they all use repetition (to which we'll dedicate an entire part of this book) to draw more than three shapes on the canvas.



Design by Luisa Pereira.



Design by Shir David.



Design by Tan Ma.

When you feel comfortable designing with basic shapes, it is time to introduce more complex shapes in your design process. We'll do this in the next chapter, where we'll be looking at a few foundational concepts from computational geometry before venturing into procedural shape generation.

EXERCISE

Design an ice cream cone in black and white with only a single occurrence of each of the basic shape functions in the code.