

Basic Commands:

```
hero.moveRight();
```

Annotations: **Object** (hero), **Function** (moveRight), **Semi-colon** (;), **Remember the period here** (arrow to .), **And the parentheses here** (arrow to parentheses).

**Object:** A character or thing that can do actions. In this case, you're controlling your hero. It's like the subject of a sentence: "The `hero` moves to the right." To control or change an **object**, put a period after it.

**Function:** An action that an object can do. You're telling the `hero` object to move right! It's like the verb of a sentence: "The hero `move`s to the `right`." Functions always need parentheses after them.

**Semi-colon:** JavaScript uses a semi-colon to tell where each statement ends.

Loop Syntax:

```
while (true) {
  hero.attack(enemy);
}
```

Annotations: **Infinite Loop** (while (true)), **Don't forget the curly braces!** (arrow to {}), **Argument** (enemy), **Closing curly brace** (}).

**Infinite Loop:** Code that repeats forever. When you want certain commands to repeat forever, use an **infinite loop**. Any commands indented under `while (true)` will repeat forever.

**Argument:** Extra information for a function. Some commands require extra information in order to run. In this example, we need to tell the `attack` function WHAT to attack.

**Curly Braces:** JavaScript uses curly braces `{ }` to tell which lines of code are inside the loop. Only lines inside the braces will repeat. Use indentation to keep code neat!

Variable Definition:

```
var enemy1 = "Kratt";
```

Annotations: **Example 1**, **Variable** (enemy1), **Assignment Operator** (=), **Value** ("Kratt").

**Variable:** A holder for a value. You can make any **value** or **object** into a **variable** by typing `var` and then giving it a name.

**Value (Example 1):** Information that is held inside a **variable**. In this case, "Kratt" is a **value** that also happens to be a **string**.

**Assignment Operator:** The equal sign `=` sets the **variable** to the **value** that comes after the sign.

Example 2

```
var enemy1 = hero.findNearestEnemy();
```

Annotation: **Value** (hero.findNearestEnemy()).

**Value (Example 2):** Even though this command looks different than the **value** in Example 1, it's doing something similar -- running the **function** and then saving the result inside the **variable**.

Variable Usage:

```
hero.attack(enemy1);
```

Annotations: **Variable** (enemy1), **Argument** (enemy1).

**Variable:** The **variable** you previously set as `enemy1` can be used as an **argument** in the `attack` function. This tells the computer to look for the **value** of the **variable** and perform the **function** on that **value**.

