

# Object-oriented Constructors in Python





# Constructor Syntax and Usage

- An object's attributes must be initialized before the object is usable
- A constructor allows you to
  1. Specify initial values of attributes upon creation of an object
  2. Require certain attributes be decided by the caller of the constructor
- A constructor is a *magic* method
  - Dunder-name is `__init__`
  - Has a first parameter named `self`
  - Return type is omitted

Before

```
a: Point = Point()
a.x = 10;
a.y = 0;
```

After

```
a: Point = Point(10, 0)
```

Defining a constructor

```
class Point:

    x: float
    y: float

    def __init__(self, x: float, y: float):
        self.x = x
        self.y = y
```

# The Semantics of a Constructor

- A constructor is a *magic* method
  - A function defined inside of a class
  - Dunder-name is `__init__`
  - Has a first parameter named `self`
  - Return type is omitted
- A class' constructor is *automagically* called each time the `Classname()` call expression is evaluated.
- "Magic" method because you do not call it directly by its name.
  - The programming language runtime calls it in the process of constructing an object.
- The `self` parameter is automatically assigned a reference to the new `Point` object on the heap.

## Defining a Constructor

```
class Point:  
  
    x: float  
    y: float  
  
    def __init__(self, x: float, y: float):  
        self.x = x  
        self.y = y
```

## Using a Class with a Constructor

```
a: Point = Point(10, 0)
```