

In the context of programming, a *function* is a named sequence of statements that performs computations operations. You can “call” the function by name. We have already seen few example of a *function call*:

```
print("hello")
```

The name of the function is `print`. Whatever you put inside parentheses is called the *argument* of the function. (The AP CSP test calls this *parameter*.) The argument is a *value, variable, or an expression* that we are *passing* into the function as input to the function. The function takes argument(s) and performs the task that it is designed to perform. The function `print`'s job is to simply display the argument on the screen.

```
print("hola") #pass a value "hola" to the print function
```

```
x = 10
```

```
print(x) #pass a variable x which has a value of 10 to the print function
```

```
x=10
```

```
y=2
```

```
print(x*2) #pass an expression x*2, that is evaluated to 20, to the print function
```

Some functions are designed to *returns* a result. In other words, a function “takes” an argument and “returns” a result. The result is called the *return value*.

You've used these types of functions before:

```
name = input("Enter your name:") #Here, input() RETURNS keyboard input
```

```
#you're assigning that to the variable name
```

```
x = 10
```

```
print( "x is " + str(x) ) #Here, str() RETURNS a string value of 10  
-> "10"
```

```
num = input("Enter a number:")
```

```
total = int(num) * 100. #Here, int() RETURNS an integer value of  
input in string data type
```

There are many other functions that comes with Python automatically but we'll just focus on following functions in AP CSP:

```
print() #display
```

```
input() #get input
```

```
int() #casting functions
```

```
str()
```

```
float()
```

```
random.randint() #generate a random number (import random is  
required)
```

```
len() #returns a length of a list (how many items are there?)
```