



# Funciones

## Capítulo 4

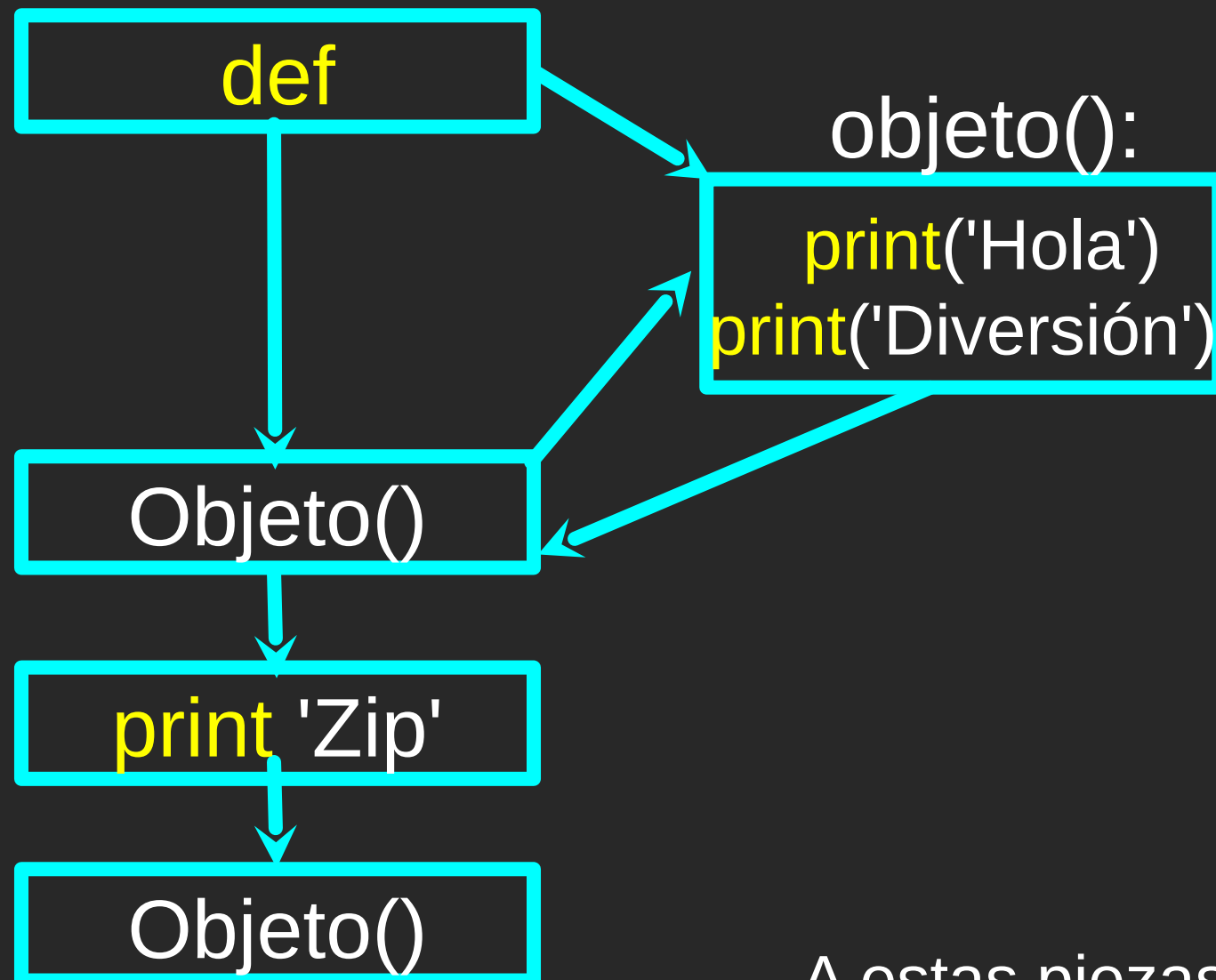


Python para Todos

[www.py4e.com](http://www.py4e.com)



# Pasos Almacenados (y reutilizados)



Programa:

```
def objeto():
    print('Hola')
```

```
print('Diversión')
```

```
objeto()
print('Zip')
objeto()
```

Resultado:

```
Hola
Diversión
Zip
Hola
Diversión
```

Green arrows point from the function calls in the code to the corresponding output lines: from the first 'objeto()' call to 'Hola' and 'Diversión', from the 'print('Zip')' call to 'Zip', and from the second 'objeto()' call to 'Hola' and 'Diversión'.

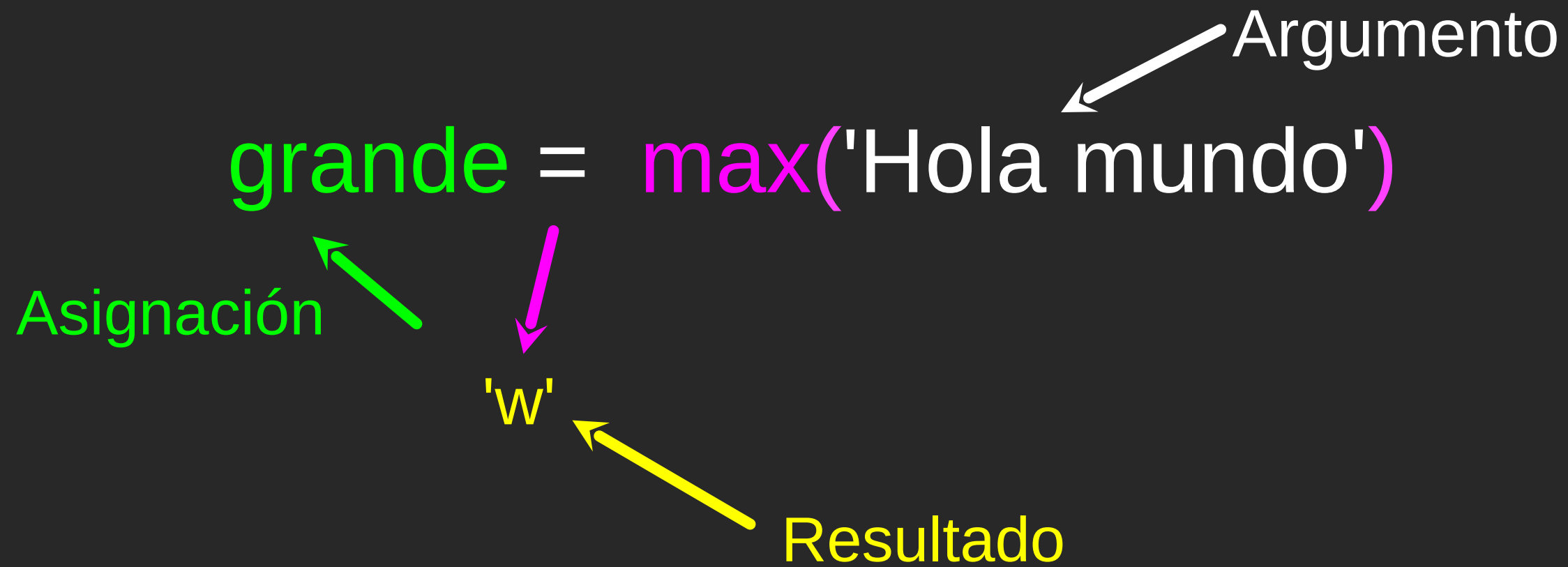
A estas piezas de códigos reutilizables las denominamos “funciones”

# Funciones de Python

- Existen dos tipos de funciones en Python.
  - **Funciones incorporadas** que se presentan como parte de Python - `print()`, `input()`, `type()`, `float()`, `int()` ...
  - **Funciones que nosotros definimos** y luego utilizamos
- Tratamos a los nombres de las funciones incorporadas como “nuevas” **palabras reservadas** (es decir, las evitamos como nombres de variables)

# Definición de la Función

- En Python una **función** es un código reutilizable que toma **argumentos**(s) como input, realiza algunos cálculos y luego devuelve uno o más resultado(s)
- Para definir una **función** utilizamos la palabra reservada **def**
- Llamamos/Invocamos a la **función** utilizando una expresión que contenga el nombre de la función, paréntesis y **argumentos**



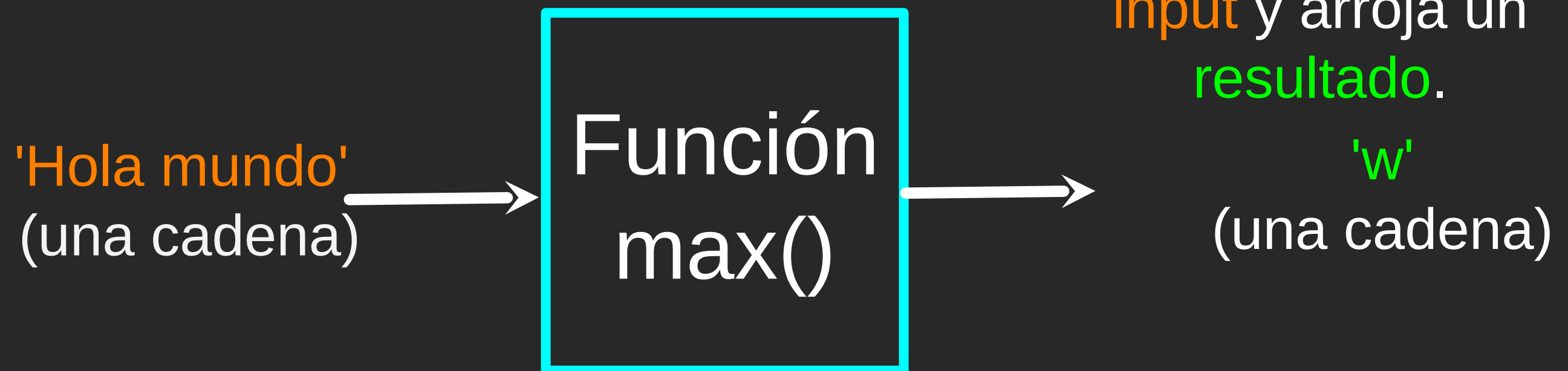
```
>>> grande = max('Hola mundo')
>>> print(grande)
w
>>> pequeño = min('Hola mundo')
>>> print(pequeño)

>>>
```

# Función Max

Una **función** es un **código almacenado** que nosotros utilizamos. Una función toma un **input** y arroja un **resultado**.

```
>>> grande = max('Hola mundo')  
>>> print(grande)  
w
```



Guido escribió este código

# Función Max

Una **función** es un **código almacenado** que nosotros utilizamos. Una función toma un **input** y arroja un **resultado**.

```
>>> grande = max('Hola mundo')
>>> print(grande)
w
```

'Hola mundo'  
(una cadena)



```
def max(inp):
    blah
    blah
    for x in inp:
        blah
        blah
```



'w'  
(una cadena)

Guido escribió este  
código

# Conversiones de Type (Tipo)

- Cuando coloca un número entero y un punto flotante en una expresión, el número entero **implícitamente** se convierte en decimal
- Puede controlar esto con las funciones incorporadas `int()` y `float()`

```
>>> print float(99) / 100
0.99
>>> i = 42
>>> type(i)
<class 'int'>
>>> f = float(i)
>>> print(f)
42.0
>>> type(f)
<class 'float'>
>>> print(1 + 2 * float(3) / 4 - 5)
-2.5
>>>
```



# Conversiones de Cadenas

- También puede usar `int()` y `float()` para convertir entre cadenas y valores enteros
- Se mostrará **error** si la cadena no contiene caracteres numéricos

```
>>> sval = '123'
>>> type(sval)
<class 'str'>
>>> print(sval + 1)
Traza de rastreo (llamada más reciente
a lo último):
  Archivo "<stdin>", línea 1, in
<module>
TypeError: cannot concatenate 'str'
and 'int'
>>> ival = int(sval)
>>> type(ival)
<class 'int'>
>>> print(ival + 1)
124
>>> nsv = 'hola bob'
>>> niv = int(nsv)
Traza de rastreo (llamada más reciente
a lo último):
  Archivo "<stdin>", línea 1, in
<module>
ValueError: invalid literal for int()
```



# Una Función Propia



## Agradecimientos / Colaboraciones



Estas diapositivas están protegidas por derechos de autor 2010-Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) de la Facultad de Información de la Universidad de Michigan y [open.umich.edu](http://open.umich.edu), y se ponen a disposición bajo licencia de Creative Commons Attribution 4.0. Por favor, conserve esta última diapositiva en todas las copias del documento para cumplir con los requisitos de atribución de la licencia. Si realiza algún cambio, siéntase libre de agregar su nombre y el de su organización a la lista de colaboradores en esta página cuando republique los materiales.

Desarrollo inicial: Charles Severance, Facultad de Información de la Universidad de Michigan

... Ingrese nuevos colaboradores y traductores aquí

...