

Fundamentos básicos del lenguaje Python

Introducción

Universidad Privada Boliviana

Hugo Condori Quispe, Ph.D.

hugo.condori@fulbrightmail.org

Introducción

Acerca de mi

Ingeniero electrónico, **Universidad Mayor de San Andres (2010)**

Maestría: Electrical and Computer Engineering, **Montana State University (2014)**

Doctorado: Electrical and Computer Engineering, **University of Utah (2018)**

Maestría: Maestria en artes liberales, “Data Science”, **Harvard University (En curso)**

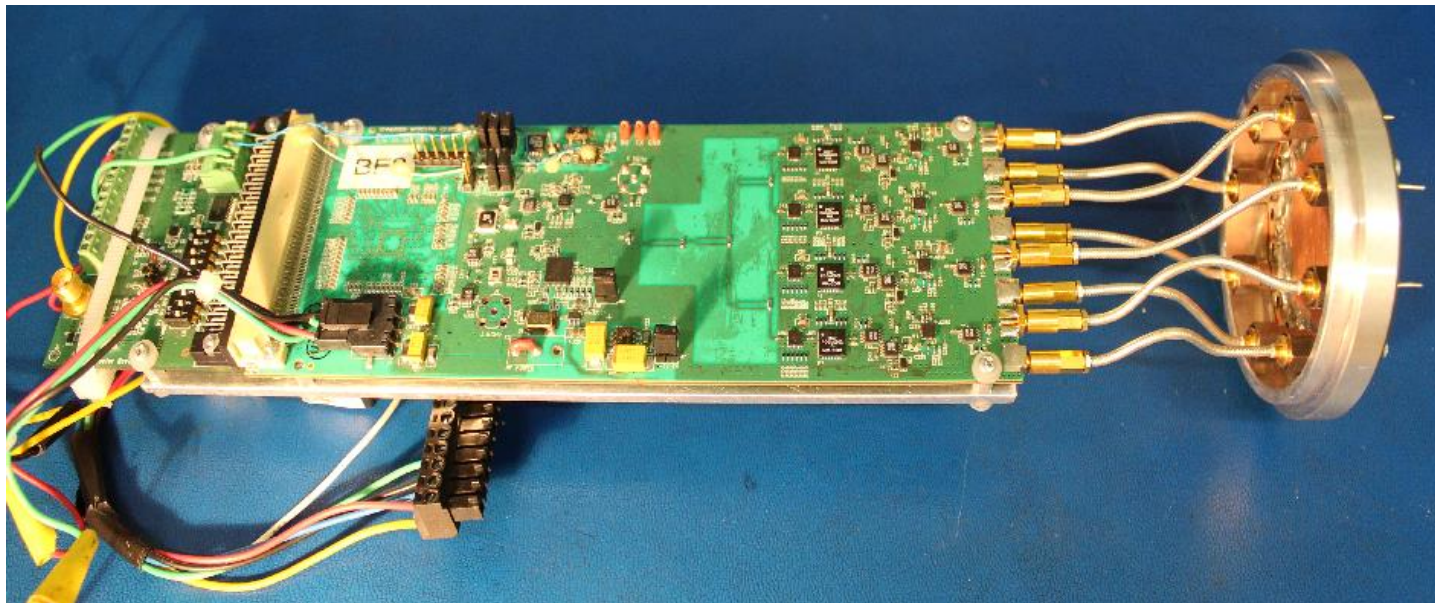


Tesis maestria: Null steering algorithms for smart antennas

Corpus ID: 58928135

Implementation of null steering algorithms in a compact analog array

[Condori Quispe](#), [Hugo Orlando](#) • Published 2014 • Computer Science

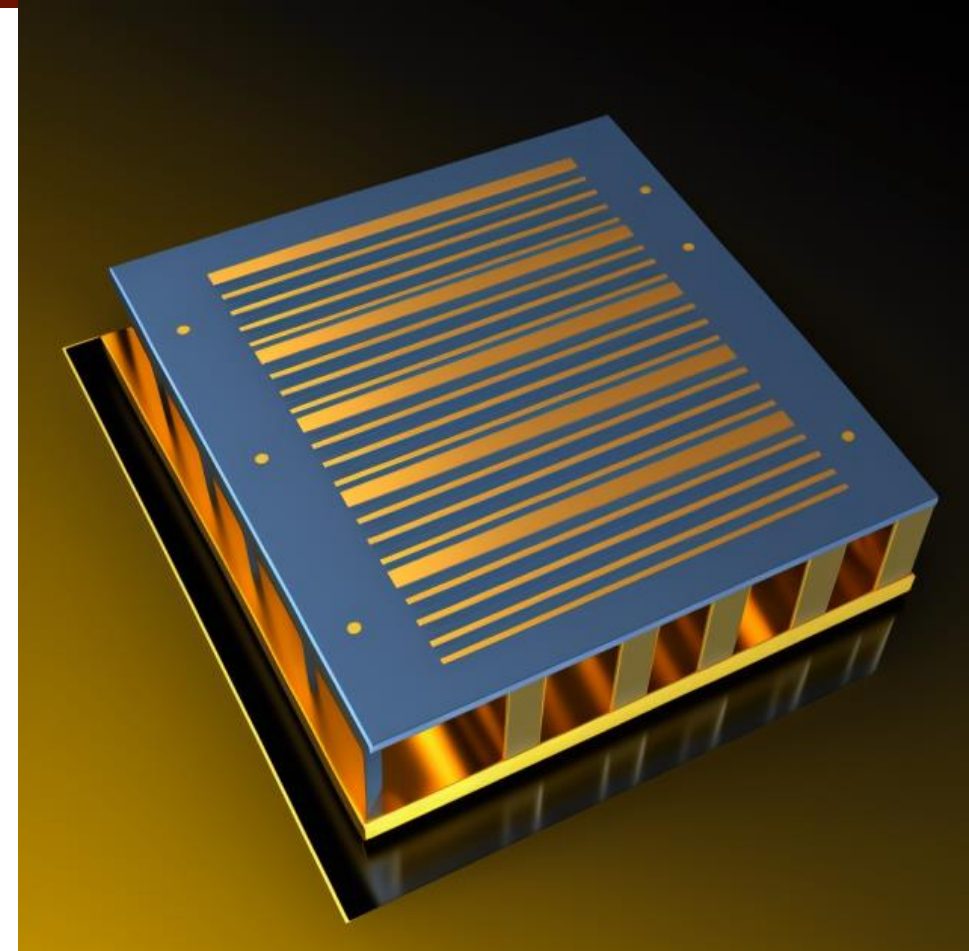


Tesis doctorado: Terahertz power amplifiers

Full Text | Dissertation or Thesis

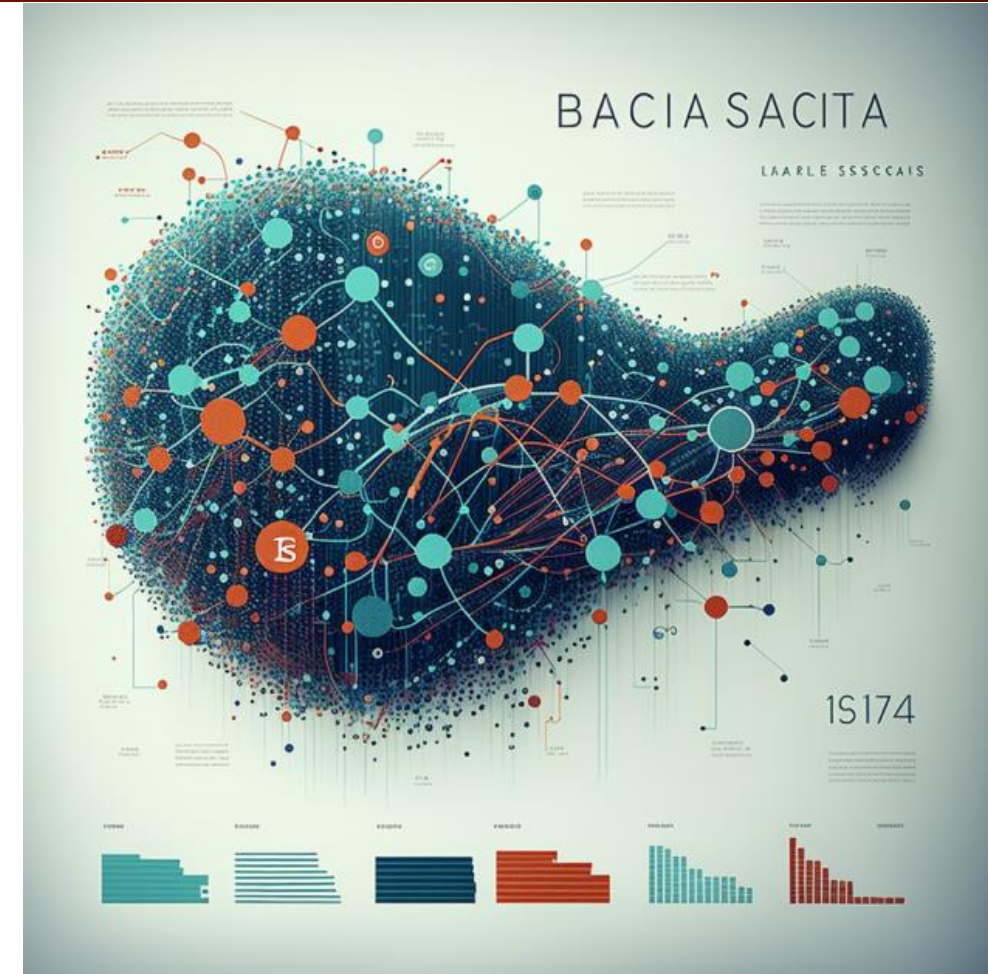
Terahertz Characterization of III-Nitride Compound Semiconductors and Novel Device Structures

Condori Quispe, Hugo Orlando. The University of Utah
ProQuest Dissertations Publishing, 2019. 13884118.



Maestría en artes liberales: Data Science

- En curso...



Harvard Extension School
HARVARD DIVISION OF CONTINUING EDUCATION

Actividad profesional



Formato

Formato del curso

Ponderación:

- Cuestionarios/actividades asincrónicas: 20%
 - A terminar el último día del módulo.
- Actividades asincrónicas: 20%
 - A entregar mediante la plataforma
- Trabajo final: 60%
 - A terminar el último día del módulo.
- Preguntas/consultas:
 - Después de clases
 - Mediante zoom previa coordinación.

Formato del curso

- Actividades asíncronas:
- Las actividades asíncronas se entregan mediante la plataforma como archivos ipynb

SESION	Fecha de entrega
--------	------------------

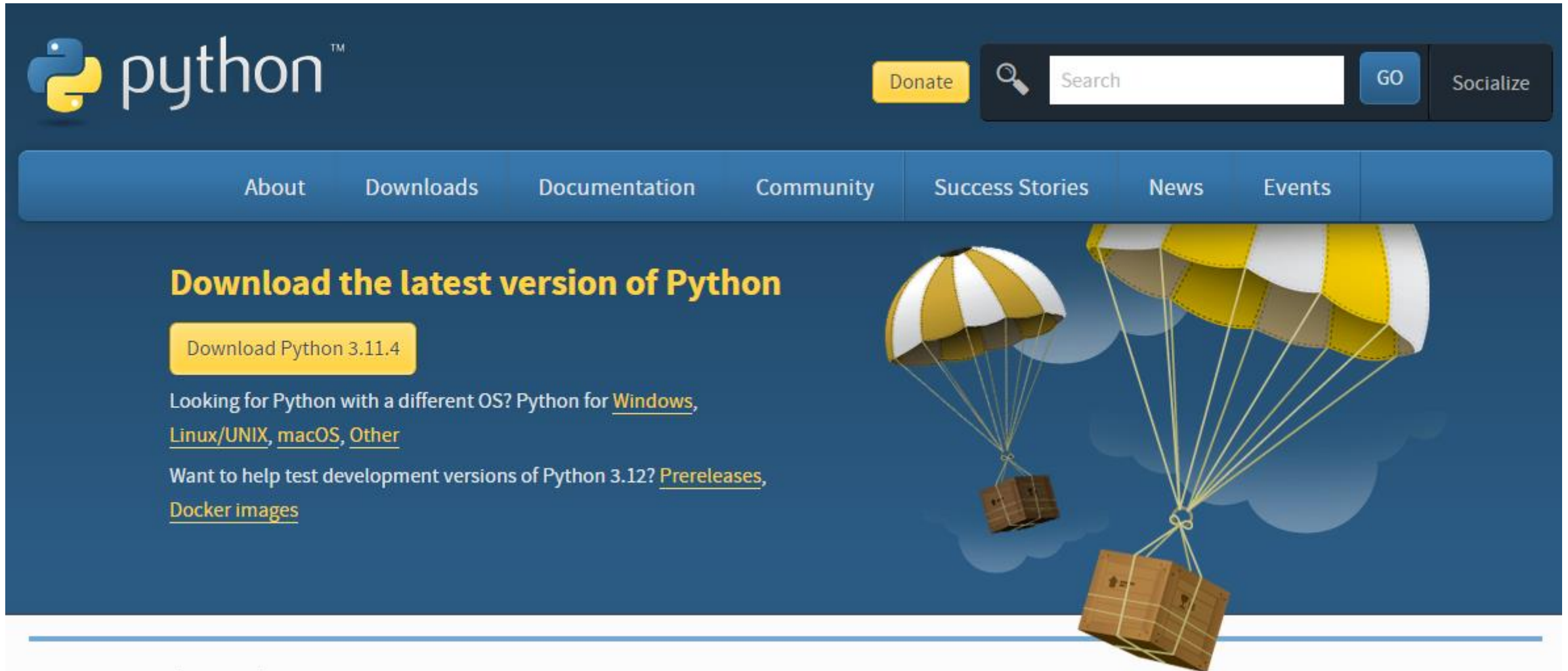
- | | |
|------------------|------------------|
| • Sesión 1: | 28 de agosto |
| • Sesión 2: | 2 de septiembre |
| • Sesión 3: | 4 de septiembre |
| • Sesión 4: | 6 de septiembre |
| • Sesión 5: | 9 de septiembre |
| • Sesión 6: | 11 de septiembre |
| • Trabajo final: | 13 de septiembre |

- Entrega fuera de plazo:
 - Entrega antes de la fecha límite: Calificación sobre 100%
 - Entrega 1 día de retraso: Calificación sobre 80%
 - Entrega >2 día de retraso: Calificación sobre 60% (hasta el 11 de septiembre)

Introducción al lenguaje Python

Instalación de Python

<https://www.python.org/downloads/>



The screenshot shows the Python.org website with a dark blue header. The Python logo is on the left, and navigation links (About, Downloads, Documentation, Community, Success Stories, News, Events) are in the center. On the right, there is a 'Donate' button, a search bar with a 'GO' button, and a 'Socialize' link. The main content area has a large yellow button that says 'Download Python 3.11.4'. Below this button, there is text linking to Python for different operating systems: 'Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)'. Further down, it says 'Want to help test development versions of Python 3.12? [Prereleases](#), [Docker images](#)'. On the right side of the main content area, there is an illustration of two parachutes with yellow and white stripes, each carrying a brown cardboard box, set against a blue sky with white clouds.

python™

Donate

Search

GO

Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version of Python

Download Python 3.11.4

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.12? [Prereleases](#), [Docker images](#)

Jupyter lab

- Para instalar jupyter lab:

JupyterLab

Install JupyterLab with `pip`:

```
pip install jupyterlab
```

- En el terminal introducir:

```
D:\UPB_DataVis>pip install jupyterlab  
WARNING: Ignoring invalid distribution  
e-packages)
```

Visual studio code

Code editing. Redefined.

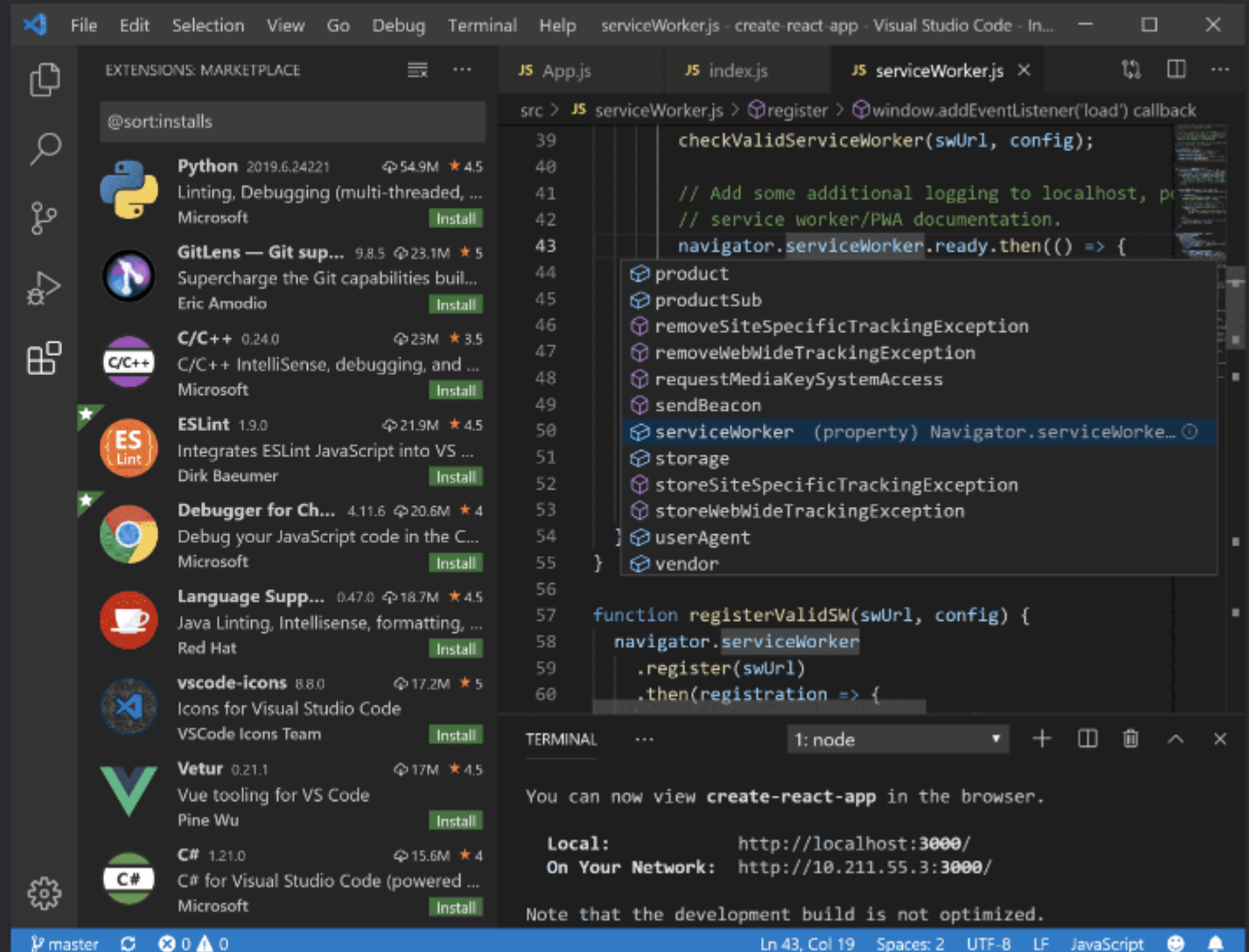
Free. Built on open source. Runs everywhere.

Download for Windows

Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its
license and privacy statement.



Python

Tipos de datos

Tipos de datos

- Integer <int>
- Float <float>
- String <str>
- Boolean <bool>

```
In [33]: type(2)
```

```
Out[33]: int
```

```
In [34]: type(2.5)
```

```
Out[34]: float
```

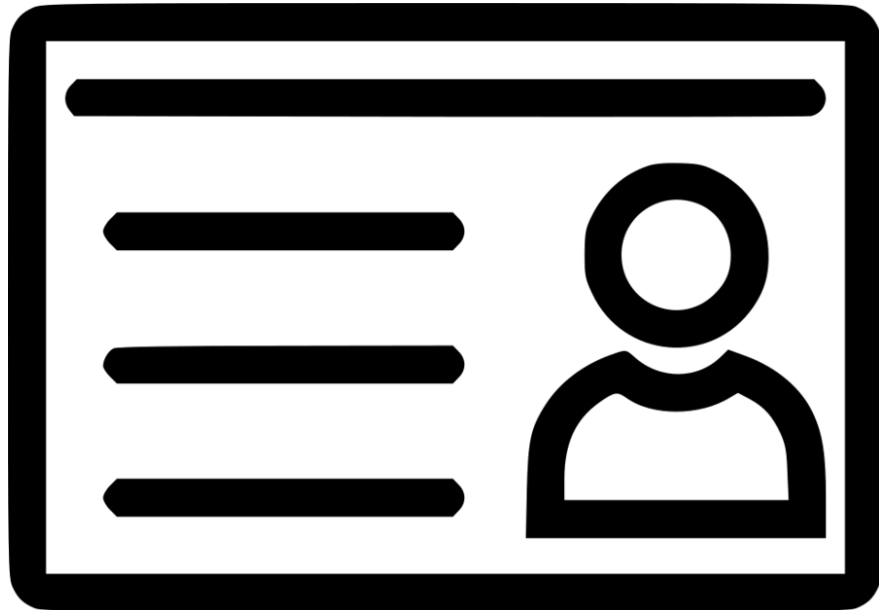
```
In [35]: type(True)
```

```
Out[35]: bool
```

```
In [36]: type('Hugo')
```

```
Out[36]: str
```


Tipos de datos



TYPE: `string`

VALUE: `'Hugo'`

ID: `54949021`

Conversión de datos

• <int> → <float>

• <float> → <int>

• <str> → <int>

• <bool> → <int>

```
In [28]: float(2)
```

```
Out[28]: 2.0
```

```
In [29]: int(2.5)
```

```
Out[29]: 2
```

```
In [30]: int('2')
```

```
Out[30]: 2
```

```
In [31]: int('hugo')
```

```
-----  
ValueError
```

Traceback

```
k (most recent call last)
```

```
<ipython-input-31-25821db0a4ee> in <module>
```

```
----> 1 int('hugo')
```

```
ValueError: invalid literal for int() with base  
10: 'pavlos'
```

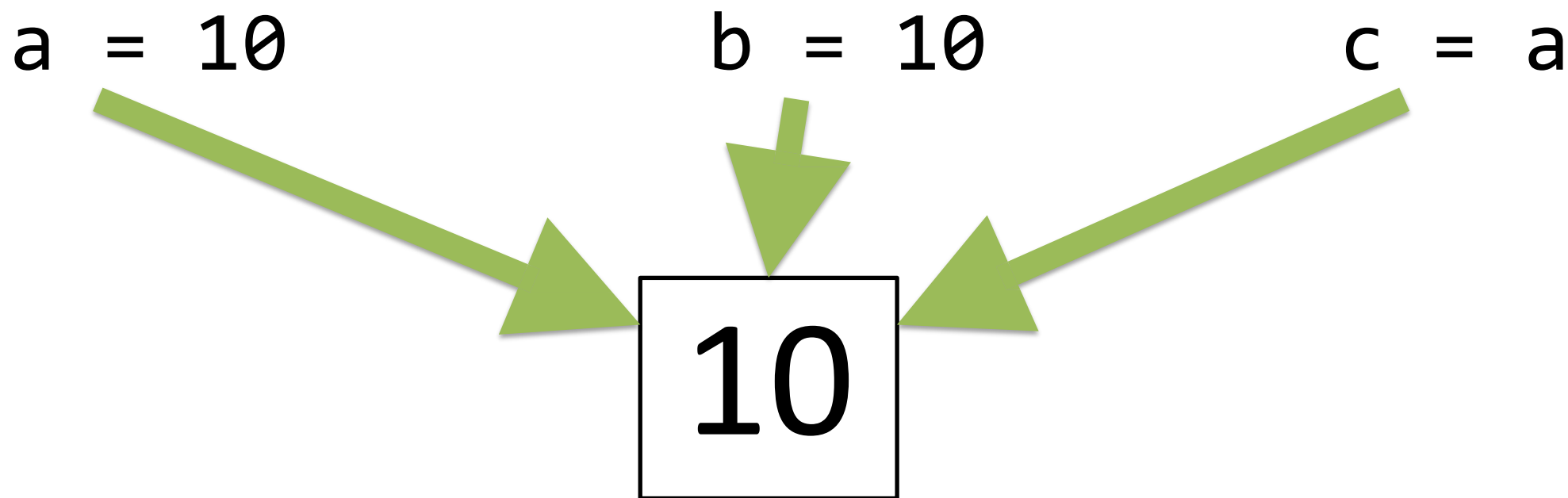
Operadores aritméticos

Operador	Significado	Ejemplo	Res.
+	Adición	$4 + 2$	6
-	Sustracción	$4 - 2$	2
*	Multiplicación	$4 * 2$	8
/	División	$4 / 2$	2
%	Operador módulos Obtiene el residuo en divisiones con enteros	$5 \% 2$	1
**	Exponente	$5^{**}2 = 5^2$	25
//	División entera	$5 // 2$ $-5 // 2$	2 -3

Variables y condiciones

Variables

Las variables se utilizan para almacenar información para ser referenciada y manipulada en un programa de computadora.



Reglas para nombrar variables

- Cada nombre de variable debe comenzar con letras del alfabeto o guion bajo (_).
- No se permiten espacios en la declaración de variables.
- No se permiten otros símbolos especiales en medio de la declaración de la variable.
- Una variable se escribe con una combinación de letras, números y caracteres especiales _ (guion bajo)

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Nombres reservados

Operadores de comparación

Operador	Significado	Ejemplo	Res.
<	Menor que	5<2	False
>	Mayor que	5>2	True
<=	Menor o igual que	5<=2	False
>=	Mayor o igual que	5>=2	True
==	Igual a	5==2	False
!=	No es igual a	5!=2	True

operador de asignación: se utiliza para asignar valores a las variables

= v / s ==

Operador para verificar si dos valores son iguales o no.

Operadores de asignación

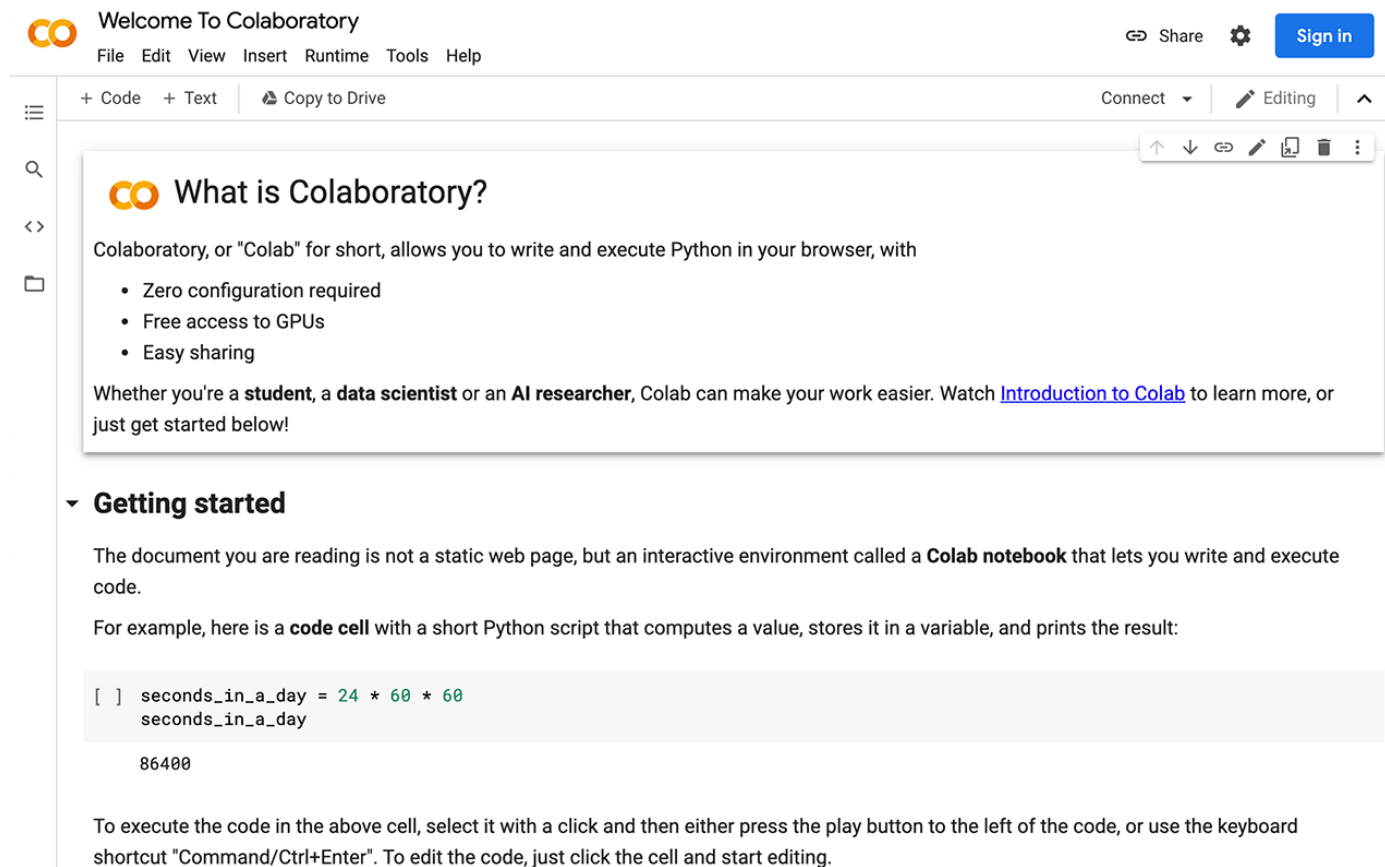
Operator	Ejemplo	Expresión Equivalente (m=15)	Res.
=	y = a+b	y = 10 + 20	30
+=	m +=10	m = m+10	25
-=	m -=10	m = m-10	5
*=	m *=10	m = m*10	150
/=	m /=10	m = m/10	1.5
%=	m %=10	m = m%10	5
=	m=2	m = m**2 or $m = m^2$	225
//=	m//=10	m = m//10	1

Operadores logicos

No olvides los paréntesis entre las condiciones.

Operador	Significado	Ejemplo	Res.
and	Operador logico “y”	(5<2) and (5>3)	False
or	Operador logico “o”	(5<2) or (5>3)	True
not	Operador logico “negacion”	not (5<2)	False

Google colab



The screenshot displays the Google Colaboratory web interface. At the top, there's a header with the Colab logo, the text 'Welcome To Colaboratory', and a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. To the right of the header are 'Share', 'Settings', and 'Sign in' buttons. Below the header, a toolbar shows '+ Code', '+ Text', and 'Copy to Drive' options. The main content area is titled 'What is Colaboratory?' and contains the following text:

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

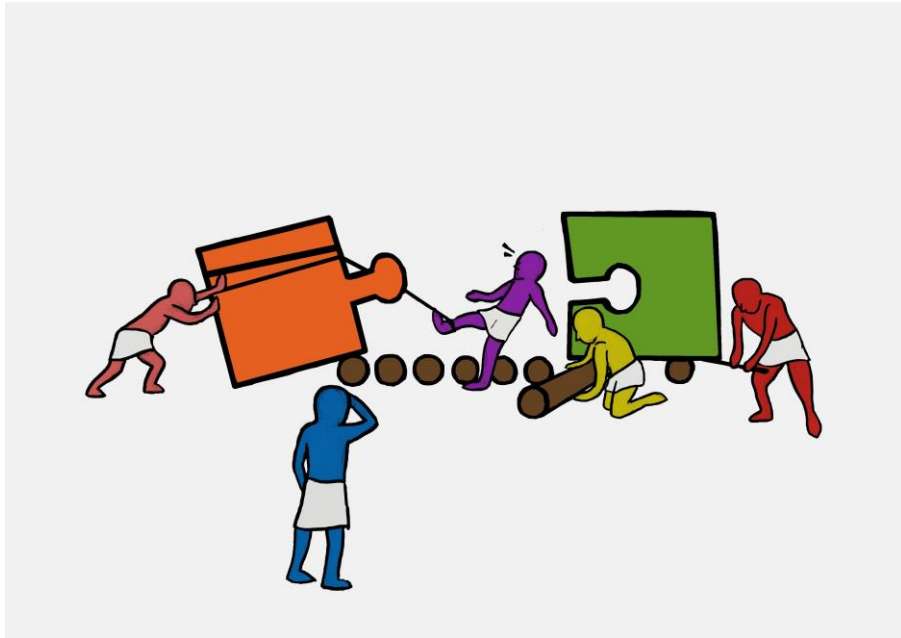
For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

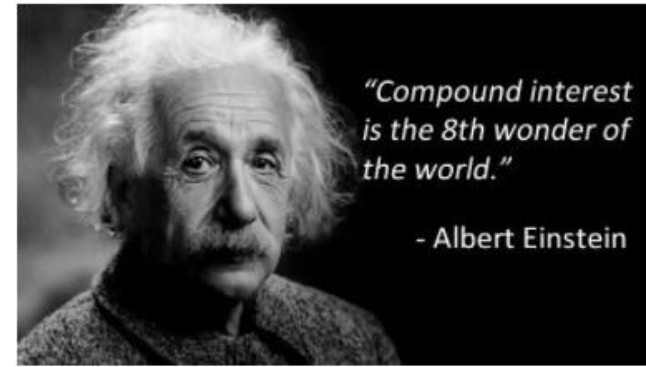
Ejercicio 1



Python como Calculadora de Ahorros 💰

¡El nuevo trabajo de ciencia de datos le ha dejado mucho dinero para invertir!

Usemos python para ver lo que puedes ganar invirtiendo en un banco.



Note: Einstein (probably) never said this.

Instrucciones:

- Siga el notebook para llenar los espacios en blanco y construya una calculadora de ahorro simple.
- Combine diferentes operadores aritméticos.
- Imprima los resultados utilizando variables, operadores y sentencias condicionales de Python.