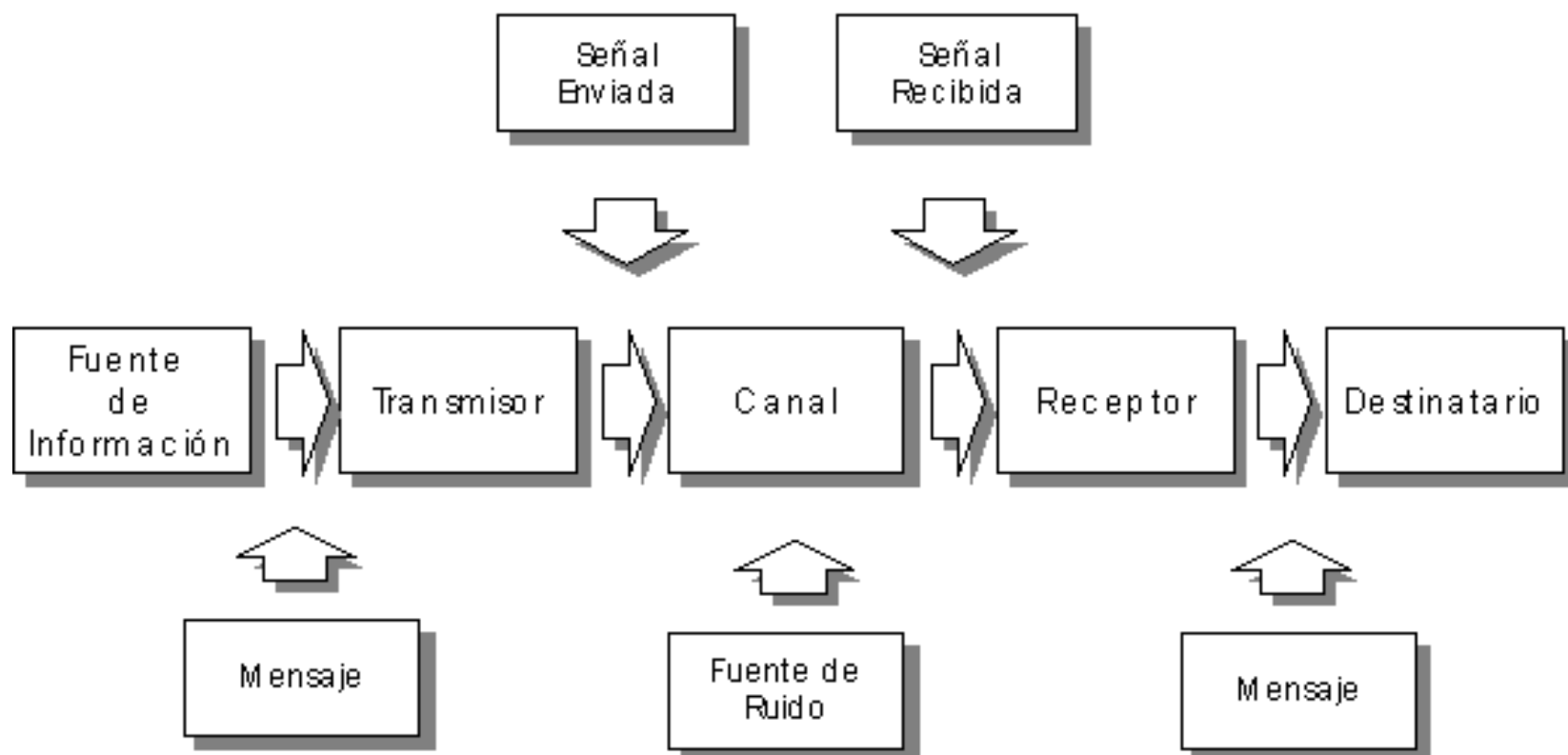


Fundamentos de Programación

Grado en Ingeniería del Software

Presentación de la asignatura

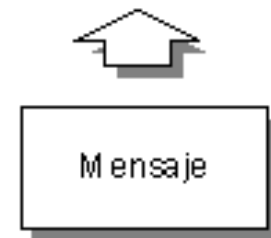
Jorge García - jorgarcia@us.es - F1.56




















Señal
Enviada

Señal
Recibida



Objetivos

- Aprender los fundamentos de la programación
- Aprender a escribir programas modulares y legibles en dos lenguajes de programación distintos: Python y Java
- Dar los primeros pasos en el desarrollo software

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.74%	+4.07%
2	1	▼	 C	13.96%	+2.13%
3	3		 Java	11.72%	+0.60%
4	4		 C++	9.76%	+2.63%
5	5		 C#	4.88%	-0.89%
6	6		 Visual Basic	4.39%	-0.22%
7	7		 JavaScript	2.82%	+0.27%
8	8		 Assembly language	2.49%	+0.07%
9	10	▲	 SQL	2.01%	+0.21%
10	9	▼	 PHP	1.68%	-0.17%
11	24	▲▲	 Objective-C	1.49%	+0.86%
12	14	▲	 Go	1.16%	+0.03%
13	20	▲▲	 Delphi/Object Pascal	1.09%	+0.32%
14	16	▲	 MATLAB	1.06%	+0.04%
15	17	▲	 Fortran	1.03%	+0.02%

Índice

- Organización
- Metodología
- Evaluación

ORGANIZACIÓN

Profesores

Grupo	Teoría	Laboratorio
1	Jorge García Gutiérrez	Jorge García Gutiérrez Cristina Rubio Escudero Belén Vega Márquez

Las tutorías de los profesores se pueden consultar en:

<http://www.lsi.us.es/personal/personal2.php?tipo=1>

Tutorías de Jorge García:

- lunes de 12:30 a 14:30,
- miércoles y viernes de 9:30 a 11:30

Clases de teoría

Grupo	Aula	C1	C2
1	A0.12	Miércoles De 12:40 a 14:30	Miércoles de 08:30 a 10:20

Clases de laboratorio

- Tres subgrupos de laboratorio:

Subgrupo 1: aula F1.31. Profesor: Jorge García

Subgrupo 2: aula F1.32. Profesora: Cristina Rubio

Subgrupo 3: aula F1.33. Profesora: Belén Vega

Grupo	Aulas	A partir del 19/10
1	A4.32 F1.30 F1.32	Lunes de 10:40 a 12:30

Herramientas de desarrollo

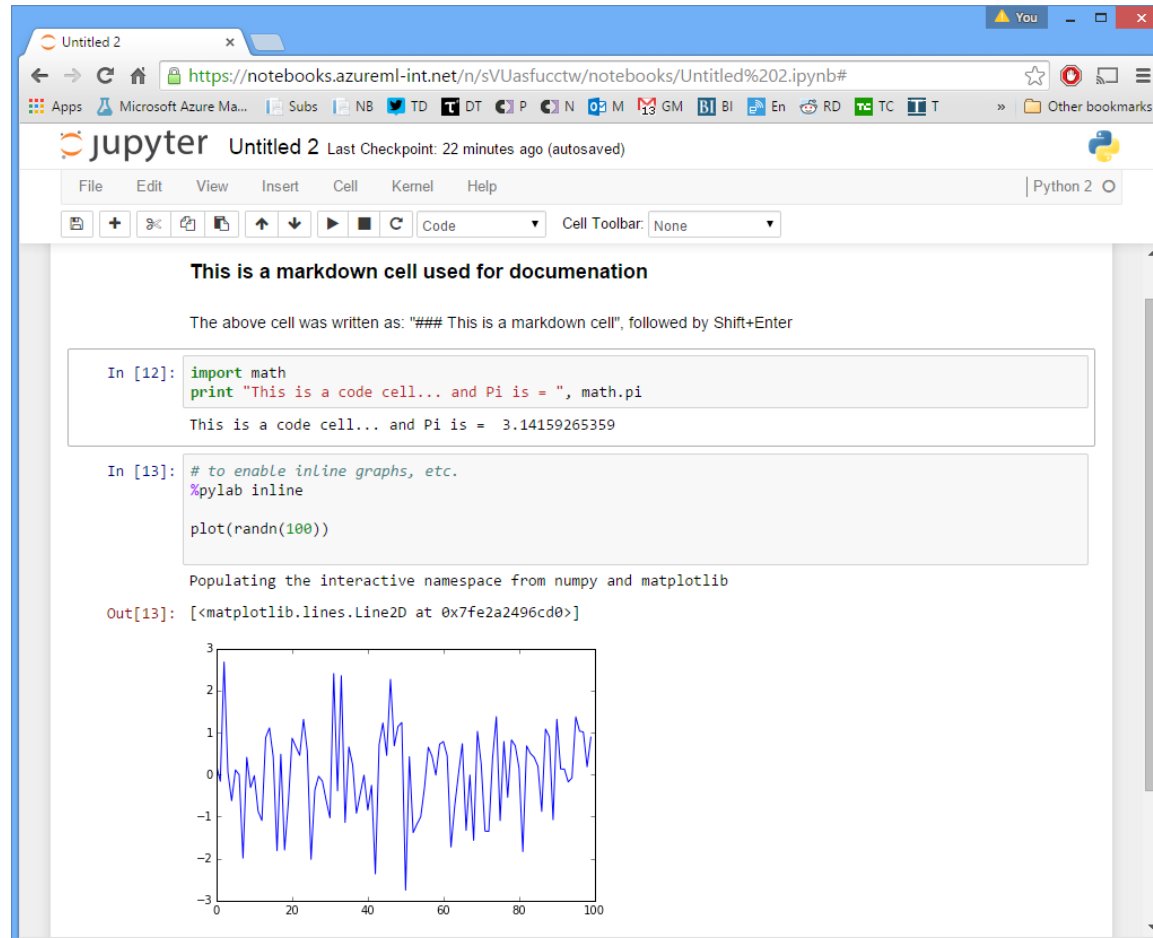


- IDE:
 - *Integrated Development Environment*
 - *Visual Studio Code*

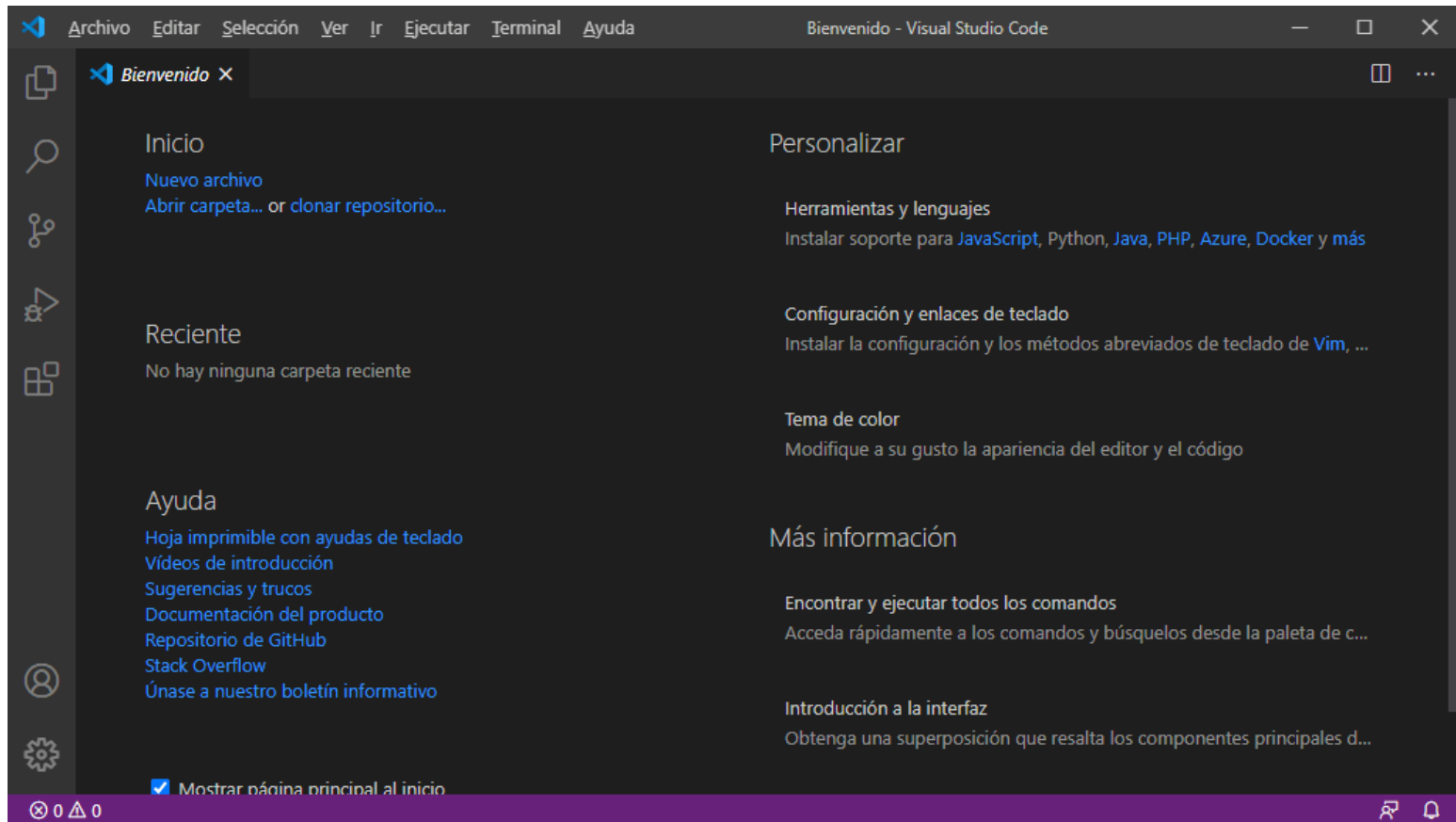


- Notebook:
 - Entorno web de computación interactiva
 - Inspirado en los notebooks de Mathematica y Sage

Los notebooks de Jupyter



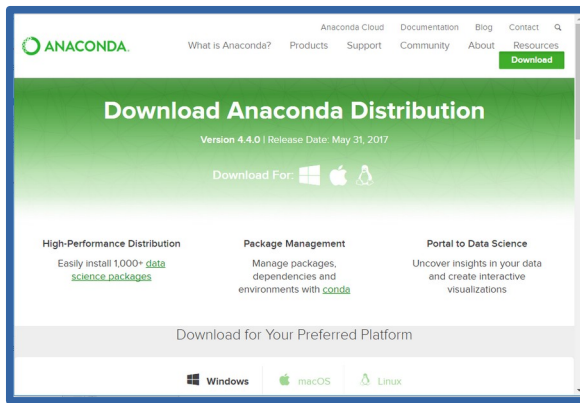
Visual Studio Code



Instalación de herramientas

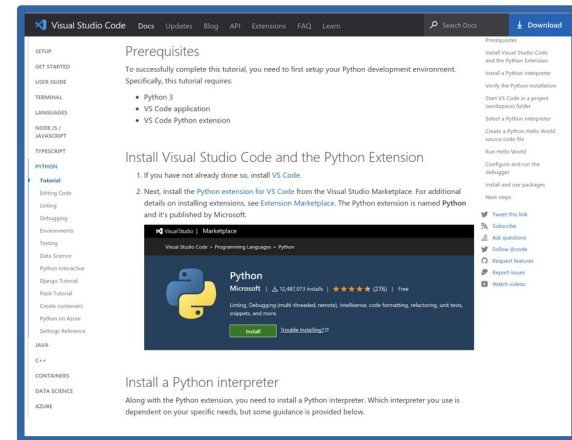
(ver la guía de instalación de herramientas de la asignatura para una lista de pasos más detallada)

1



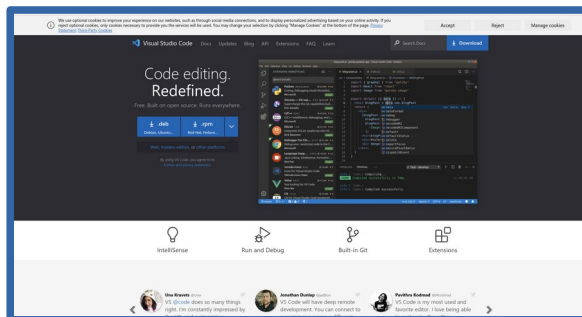
<https://www.anaconda.com/download/>

3



<https://code.visualstudio.com/docs/python/python-tutorial>

2



<https://code.visualstudio.com>

Curso 0 de Fundamentos de Programación

The screenshot shows a web browser window displaying the Blackboard LMS interface. The address bar shows the URL: https://ev.us.es/webapps/portal/frameset.jsp?tab_tab_group_id=_2_1&url=%2Fwebapps%2Fblackboard. The page header includes the Blackboard logo and the text "ENSEÑANZA VIRTUAL". The navigation bar shows "Mi institución", "Soporte de Enseñanza Virtual", "Cursos", and "Servicios". The main content area is titled "Presentación" and features a sidebar with a list of courses: "Curso 0 de la ETSII", "Física", "Matemáticas", "Programación", "Mis calificaciones", "Mis logros", and "Ayuda". The main content area displays the "Curso 0" section, which includes a welcome message, a list of topics, and a description of the evaluation process. The topics listed are: "Programación", "Matemáticas", and "Física". The evaluation process is described as a "prueba de evaluación" that can be taken multiple times, with the best score being recorded. The page footer includes the Blackboard logo and copyright information: "© 1997-2017 Blackboard Inc. Todos los derechos reservados. Número de patente 7.493.396 y 7.558.853 de Estados Unidos. Resto de patentes en trámite. Información sobre accesibilidad Detalles de instalación".

Blackboard LMS Interface - Curso 0 de la ETSII

Header: Blackboard LMS, ENSEÑANZA VIRTUAL, Mi institución, Soporte de Enseñanza Virtual, Cursos, Servicios, Go To Teacher View.

Left Sidebar:

- Curso 0 de la ETSII
 - Presentación
 - Física
 - Matemáticas
 - Programación
 - Mis calificaciones
 - Mis logros
 - Ayuda

Main Content Area:

Presentación

Curso 0

Bienvenido al **Curso 0 de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla**. Este curso pretende servir de ayuda a los alumnos de nuevo ingreso, sirviéndoles para recordar conocimientos ya adquiridos en sus estudios de Bachillerato y que van a necesitar en las asignaturas de primer curso de Matemáticas y Física. En el caso de Programación, no se presuponen conocimientos previos, y el curso está pensado para obtener unos conocimientos elementales como apoyo a la asignatura Fundamentos de Programación.

Los contenidos disponibles en este curso son los siguientes:

- Programación:** Contenidos correspondientes a la asignatura de Fundamentos de Programación. Consta de cinco temas, que abarcan desde los conceptos básicos de programación hasta una breve introducción al lenguaje Java.
- Matemáticas:** Contenidos correspondientes a las asignaturas de Matemáticas. Consta de tres partes: Aritmética, Funciones y Matrices.
- Física:** Contenidos correspondientes a la asignatura de Fundamentos Físicos de la Informática.

Al final de cada uno de estos contenidos encontrarás una **prueba de evaluación**. Puedes realizar las pruebas tantas veces como quieras, y se te guardará la mejor nota que hayas obtenido. Si superas los 7 puntos en una prueba, obtendrás automáticamente una insignia como reconocimiento. Si consigues las tres insignias, obtendrás una insignia final por haber superado el Curso 0.

Course Topics:

- Física
- Matemáticas
- Programación

Footer: Blackboard logo, © 1997-2017 Blackboard Inc. Todos los derechos reservados. Número de patente 7.493.396 y 7.558.853 de Estados Unidos. Resto de patentes en trámite. Información sobre accesibilidad Detalles de instalación.

[BasiliumLearn](#)

[https://moodle.uned.es/webapps/porta/?module=portafolios_group&courseid=421&webapp=527&block=1](#)

[Buscar](#)

[Inicio](#)
[Inicio](#)
[Inicio](#)
[Inicio](#)
[Inicio](#)

[Inicio](#)
[Inicio](#)
[Inicio](#)
[Inicio](#)
[Inicio](#)

ENSEÑANZA VIRTUAL

[Mi institución](#)
[Sistema de gestión de contenidos](#)
[Cursos](#)
[Servicios](#)

[Presentación](#)
[Programación](#)
[Temas 1. Conceptos de Programación](#)
[Go To Teacher View](#)

Curso de la ETSI

[Presentación](#)
[Física](#)
[Matemáticas](#)
[Programación](#)
[No calificaciones](#)
[Mis logs](#)
[Ayuda](#)

Temas 1. Conceptos de Programación

Conceptos de Programación

[1. Concepto de programa](#)
[2. Máquinas programables](#)
[3. Lenguajes de programación](#)
[4. Ejercicios de programación](#)
[5. La jerarquía del software](#)
[6. Tipos de programas](#)
[7. Contadores y variables](#)

Concepto de programa

Un algoritmo es un conjunto de instrucciones que permiten realizar una tarea. Hay que seguirlos todos y en el orden correcto para que la tarea se realice correctamente.

En nuestra vida diaria utilizamos algoritmos muy a menudo. Cuando nos vestimos, hacemos la maleta, preparamos una receta de cocina o cambiamos una rueda del coche, estamos aplicando un algoritmo. Veamos por ejemplo un algoritmo básico para tomar una fotografía:

1. Enfoocar la cámara
2. Quitar la tapa del objetivo
3. Seleccionar ISO
4. Seleccionar apertura del diafragma y velocidad de disparo
5. Encuadrar la escena
6. Enfocar
7. Disparar

Está claro que si no quitamos la tapa del objetivo, o si hacemos disparo de disparar, no vamos a obtener una buena foto. Vamos pues que es importante seguir todos los pasos y hacerlo en el orden que nos indica el algoritmo.

Curso de Programación (5 temas)

Blackboard Learn

<https://www.blackboard.com/learn/jsp/course/course.do?courseId=1037&webpage%3Faction=...>

ENSEÑANZA VIRTUAL

[Mi institución](#)
[Sopos de voz](#)
[Cursos](#)
[Servicios](#)

[Presentación](#)
[Programación](#)
[Realizar pruebas](#)
[Evaluación del tema 1 de Programación](#)

Go To Teacher View

Curso 0 de la ETSII

Presentación

Física

Matemáticas

Programación

Mis Calificaciones

Mis Tipos

Ayuda

Realizar prueba: Evaluación del tema 1 de Programación

Descripción

Prueba de evaluación del tema 1 de Programación, "Conceptos de Programación".

Esta prueba contiene 10 preguntas, cada una de las cuales vale 1 punto. En cada pregunta, las respuestas correctas suman puntos y las respuestas incorrectas restan puntos, aunque la puntuación de la pregunta nunca será inferior a 0 puntos.

Esta prueba tiene un límite de tiempo de 30 minutos. La prueba se guardará y se enviará automáticamente cuando el tiempo de espera.

Se mostrará una advertencia cuando falte la mitad del tiempo, 5 minutos, 1 minuto y 30 segundos.

Intentos múltiples

Esta prueba permite varias intentos.

Esta prueba no permite volver atrás. No es posible modificar la respuesta después del envío.

Tiempo restante: 9 minutos, 33 segundos.

Estado de finalización de la pregunta:

Pasa a otra pregunta. La impedirá realizar cambios a esta respuesta.

Pregunta 4

¿Cuál de los siguientes lenguajes de programación crees que es más cercano al lenguaje que utilizan las personas?

☐ Python C

☐ Python Java

☐ Python máquina

Pasa a otra pregunta. La impedirá realizar cambios a esta respuesta.

Puntos Guardar respuesta

Puntos Guardar respuesta

Autoevaluación para cada tema

Bibliografía

- Libros:

- S. Chazallet. Python 3. Los fundamentos del lenguaje (2ª edición). Ediciones ENI. 2016 [BIBLIOTECA]
- A.E. Downey. Think Python 2e. O'Reilly. 2015
[WEB: <http://greenteapress.com/wp/think-python-2e/>]
- C. Dierbach. Introduction to Computer Science Using Python: A Computational Problem-Solving Focus. Wiley. 2012.
- J.V. Guttag. Introduction to Computation and Programming using Python (second edition). The MIT Press. 2015.
- M. Lutz. Learning Python. O'Reilly. 2013
- C.R. Severance. Python for Everybody. Exploring Data Using Python 3. 2016

[WEB: <http://www.pythonlearn.com/>]

- Tutoriales online:

- Tutorials Point [WEB: <https://www.tutorialspoint.com/python3/>]



stackoverflow to the rescue!

stackoverflow Questions Developer Jobs Tags Users Search... ? Log In Sign Up

Stack Overflow es una comunidad de 7.8 millones de programadores como tú que se ayudan mutuamente. Únete a ellos, sólo te llevará un minuto: Registrarse

Reading a CSV file using Python Ask Question

please tell me what's the problem in this code it's giving an error

```
import csv
with open('some.csv', 'rb') as f:
    reader = csv.reader(f)
    for row in reader:
        print row
```

python csv

share improve this question

edited Apr 26 '11 at 12:08 martineau 51.3k 6 70 126

asked Apr 26 '11 at 9:39 atul 50 1 1 6

asked 6 years, 5 months ago
viewed 14,105 times
active 6 years, 5 months ago

what is the error ur getting? – Srinivas Reddy Thatiparthi Apr 26 '11 at 9:40

invalid systax at 2nd line – atul Apr 26 '11 at 9:43

1 The code looks fine to me. Without the error, or the contents of 'some.csv' it is difficult to help. – juanchopanza Apr 26 '11 at 9:43

with open('some.csv', 'rb') as f showing syntax error as invalid syntax – atul Apr 26 '11 at 9:44

7 sheesh, guys, some patience for a new user. If you don't have anything constructive to say, leave your jokes for the pub – Eli Bendersky Apr 26 '11 at 9:52

show 4 more comments

2 Answers active oldest votes

¿No encuentras la respuesta? Pregunta en Stack Overflow en español.

Which version of Python are you using?

12 The with statement is new in 2.6 - if you're using 2.5 you need from __future__ import

BLOG

Podcast #116 – What is Technology? Do We Even Know?

What Do Software Developers Use in Germany?

Want a python job?

Python Developer - Web Scraping

Scrapinghub No office location

REMOTE

python scrapy

Data Science Consultant (m/w)

Comma Soft AG Bonn, Deutschland

€50K - €90K REMOTE

python numpy

Python Developer - Build the smartest way to buy a car (Telecommute - full time)

Unhaggle Inc. No office location

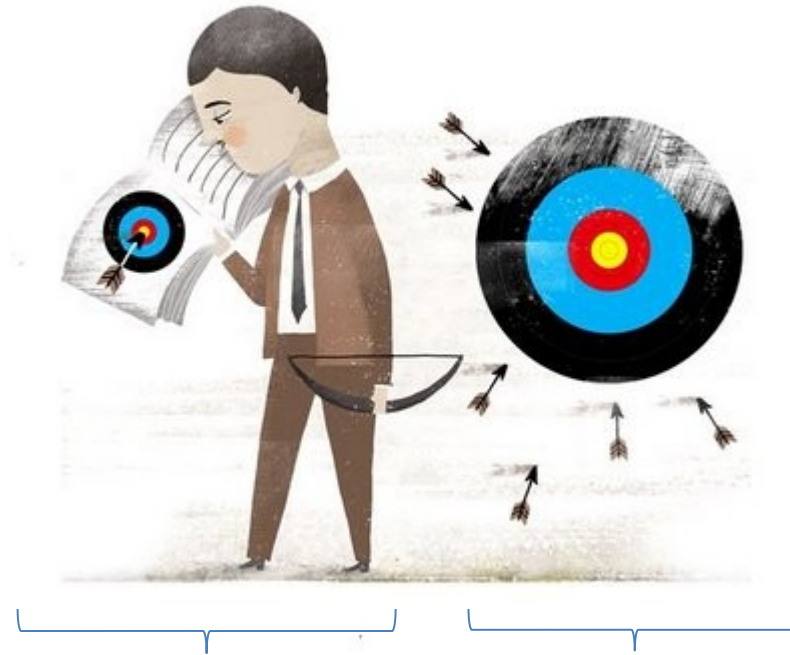
REMOTE

python django

Senior Back-end Developer (Remote)

METODOLOGÍA

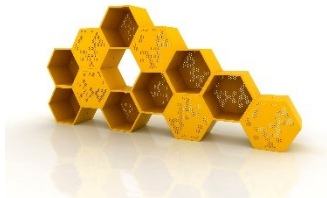
Propuesta de distribución de tiempo de estudio



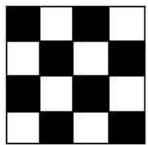
20% a
apuntes,
documentación, ...

80% a escribir
programas y probarlos

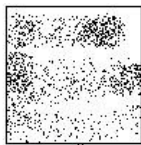
Autoevaluación: ¿cómo sé que un ejercicio está bien resuelto?



Es modular: una buena descomposición en funciones simples

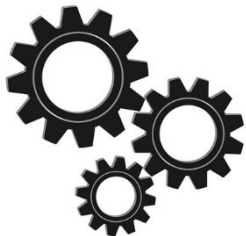


Legible,
Calming



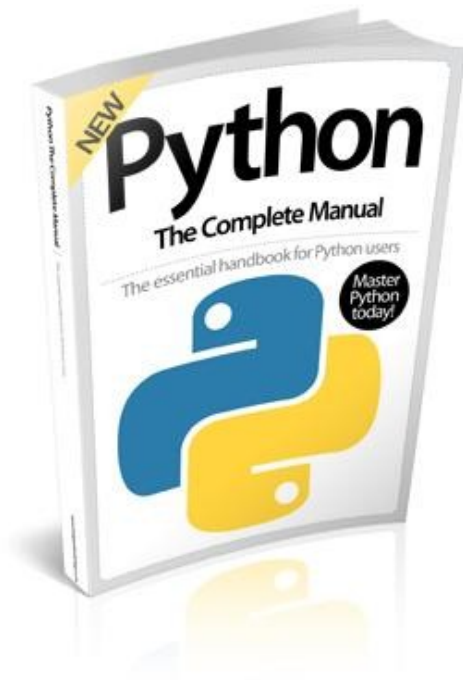
Illegible,
Anxiety-Provoking

Es legible: la solución es fácil de comprender y explicar



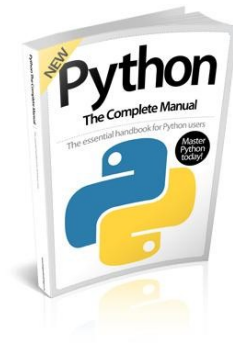
Y además, **funciona!!!**

Dos formas de aprender a programar





- Será la base de nuestro aprendizaje
- Completaremos proyectos que:
 - Sean interesantes
 - Sea fácil comprender su alcance
 - Usen diferentes elementos de Python
 - Tengan un tamaño razonable

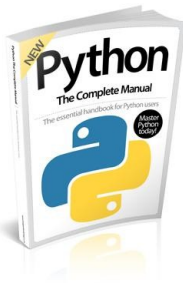


- Tendrá un papel secundario en nuestra metodología:
 - Sesiones de teoría para organizar conceptos
 - Cierre de bloques vinculados a la evaluación

Tipos de contenidos

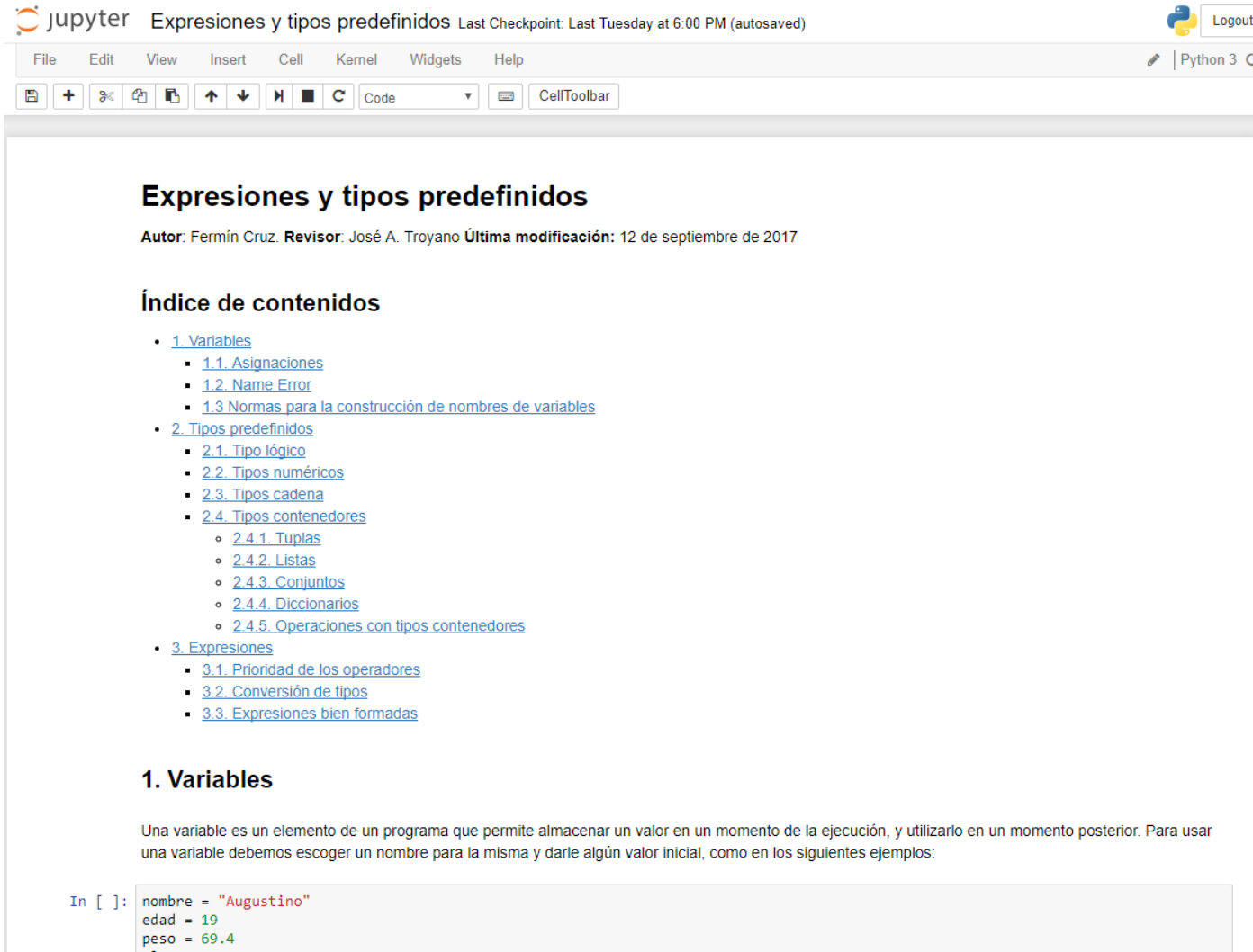


- Notebooks de ejercicios
- Proyectos de laboratorio



- Notebooks de teoría

Notebooks de teoría



The screenshot shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text 'Expresiones y tipos predefinidos' and 'Last Checkpoint: Last Tuesday at 6:00 PM (autosaved)'. On the right, there is a 'Logout' button. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar is a 'Python 3' indicator. Below the menu bar is a toolbar with icons for file operations, a dropdown menu set to 'Code', and a 'CellToolbar' button. The main content area has a title 'Expresiones y tipos predefinidos' and a subtitle 'Autor: Fermín Cruz. Revisor: José A. Troyano Última modificación: 12 de septiembre de 2017'. Below the subtitle is a section 'Índice de contenidos' with a list of links: '1. Variables' (with sub-links '1.1. Asignaciones', '1.2. Name Error', '1.3 Normas para la construcción de nombres de variables'), '2. Tipos predefinidos' (with sub-links '2.1. Tipo lógico', '2.2. Tipos numéricos', '2.3. Tipos cadena', '2.4. Tipos contenedores' which further branches into '2.4.1. Tuplas', '2.4.2. Listas', '2.4.3. Conjuntos', '2.4.4. Diccionarios', and '2.4.5. Operaciones con tipos contenedores'), and '3. Expresiones' (with sub-links '3.1. Prioridad de los operadores', '3.2. Conversión de tipos', '3.3. Expresiones bien formadas'). Below the index is a section '1. Variables' with a paragraph explaining that a variable is an element of a program that allows storing a value at one point in time and using it later. It states that to use a variable, one must choose a name and give it an initial value, as in the following examples. At the bottom, a code cell is shown with the input 'In []:' followed by three lines of code: 'nombre = "Augustino"', 'edad = 19', and 'peso = 69.4'.

jupyter Expresiones y tipos predefinidos Last Checkpoint: Last Tuesday at 6:00 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Python 3

Code CellToolbar

Expresiones y tipos predefinidos

Autor: Fermín Cruz. Revisor: José A. Troyano Última modificación: 12 de septiembre de 2017

Índice de contenidos

- [1. Variables](#)
 - [1.1. Asignaciones](#)
 - [1.2. Name Error](#)
 - [1.3 Normas para la construcción de nombres de variables](#)
- [2. Tipos predefinidos](#)
 - [2.1. Tipo lógico](#)
 - [2.2. Tipos numéricos](#)
 - [2.3. Tipos cadena](#)
 - [2.4. Tipos contenedores](#)
 - [2.4.1. Tuplas](#)
 - [2.4.2. Listas](#)
 - [2.4.3. Conjuntos](#)
 - [2.4.4. Diccionarios](#)
 - [2.4.5. Operaciones con tipos contenedores](#)
- [3. Expresiones](#)
 - [3.1. Prioridad de los operadores](#)
 - [3.2. Conversión de tipos](#)
 - [3.3. Expresiones bien formadas](#)

1. Variables



Una variable es un elemento de un programa que permite almacenar un valor en un momento de la ejecución, y utilizarlo en un momento posterior. Para usar una variable debemos escoger un nombre para la misma y darle algún valor inicial, como en los siguientes ejemplos:

```
In [ ]: nombre = "Augustino"
        edad = 19
        peso = 69.4
        altura = 1.70
```








Bloques teóricos

1. Introducción a Python
2. Expresiones, tipos predefinidos, entrada/salida
3. Instrucciones condicionales y bucles
4. Funciones
5. Secuencias, listas y tuplas
6. Diccionarios y conjuntos

Notebooks de ejercicios

 jupyter FutElo (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Python 3

       Code CellToolbar

Ejercicio: Ranking Elo aplicado a resultados de fútbol

Autor: José A. Troyano. **Revisor:** Fermín Cruz. **Última revisión:** 13/09/2017

El sistema de puntuación Elo es un método estadístico para calcular una clasificación entre competidores que no necesariamente se han enfrentado todos entre sí. Empezó a usarse en ajedrez, aunque su uso se está generalizando a otras competiciones. Fue desarrollado por el físico estadounidense de origen húngaro Árpád Élő (1903-1992), y a él debe su nombre.

Tradicionalmente el ranking Elo ha sido sinónimo de la clasificación mundial de ajedrez ([ranking FIDE](#)) y más recientemente otras federaciones deportivas han adoptado un sistema basado en Elo para elaborar sus rankings. Por ejemplo, la FIFA usa Elo para elaborar el ranking de selecciones de [fútbol femenino](#), aunque para el masculino sigue usando un método más convencional basado principalmente en el número de partidos ganados recientemente.

En este ejercicio vamos a utilizar los resultados de partidos de fútbol de primera y segunda división españolas para calcular a partir de ellos un ranking de equipos basado en el sistema Elo. Los datos utilizados ha sido recuperados de la web de resultados históricos de [LaLiga](#) y han sido adaptados para representarlos en un formato [CSV](#) que resulta muy fácil de manejar desde Python.

1. Carga de datos

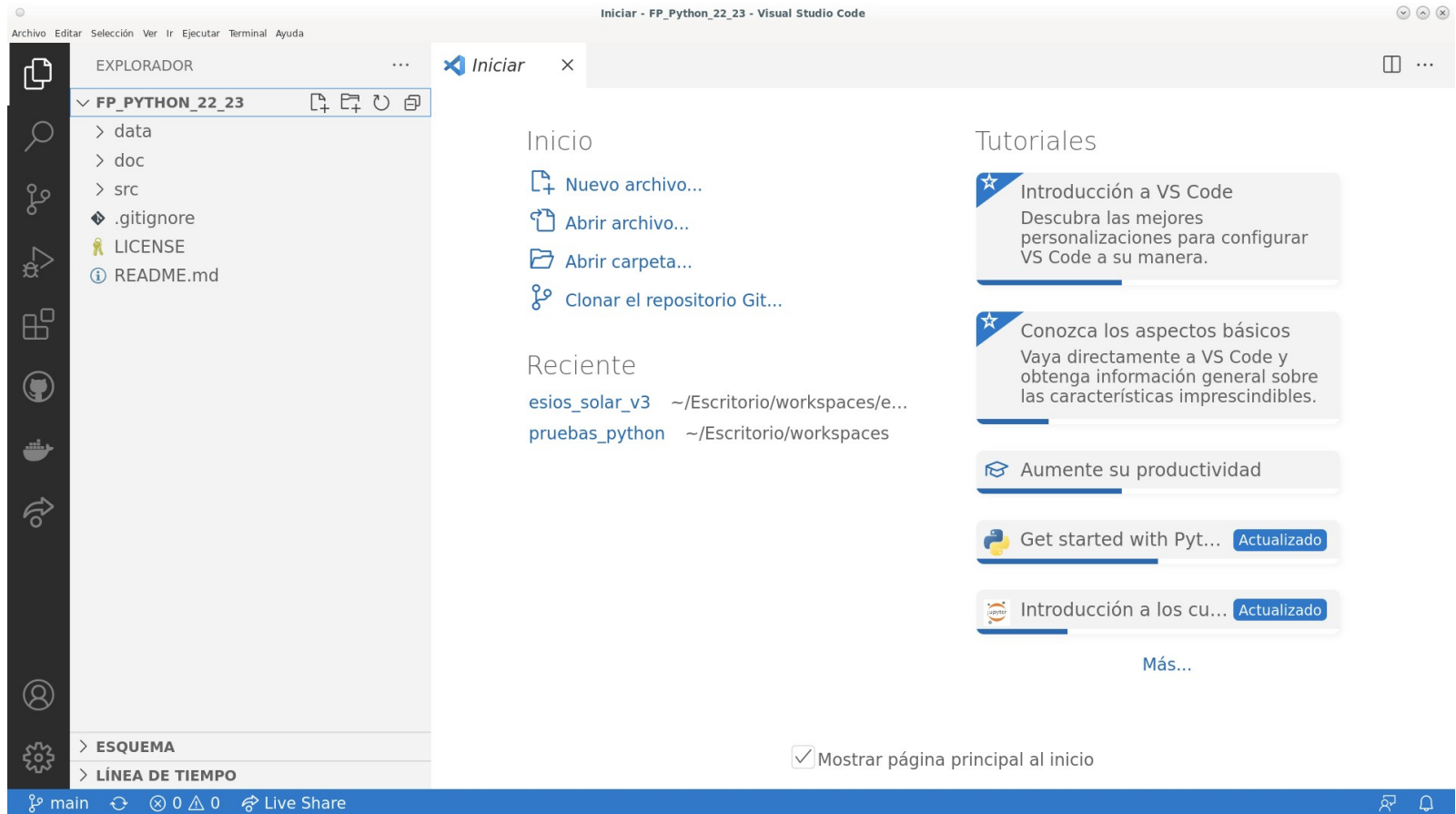
Se dispone de los resultados de primera y segunda división desde la temporada 00-01 hasta la temporada 15-16. Los datos están organizados en dos carpetas (Primera y Segunda), y dentro de cada una de ellas hay un fichero CSV para cada temporada. Cada línea de estos ficheros se corresponde con un partido, y en ella se incluyen cinco datos:

- Fecha del partido
- Equipo que juega en casa
- Goles del equipo de casa
- Equipo visitante
- Goles del equipo visitante

Estan serían, por ejemplo, las primeras líneas del fichero de la temporada 15-16 en primera división:


```
21/08/2015,Málaga CF,0,Sevilla FC,0
21/08/2015,Atlético de Madrid,1,UD Las Palmas,0
21/08/2015,RC Deportivo,0,Real Sociedad,0
21/08/2015,RCD Espanyol,1,Getafe CF,0
```


Proyectos de laboratorio




Material disponible en GitHub

<https://github.com/fupus>

 Features Business Explore Marketplace Pricing [Sign in](#) or [Sign up](#)






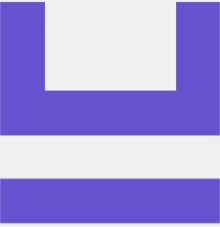
Create your own GitHub profile

Sign up for your own profile on GitHub, the best place to host code, manage projects, and build software alongside 28 million developers.

[Sign up](#)



[Dismiss](#)



fupus

Material de la asignatura "Fundamentos de Programación" impartida por el Departamento de Lenguajes y Sistemas Informáticos (Universidad de Sevilla)


[Block or report user](#)

Overview Repositories **3** Stars **0** Followers **5** Following **0**

Popular repositories


[notebooks-teoria](#)

Notebooks Jupyter de introducción a Python

 Jupyter Notebook ★ 1 🍴 2


[notebooks-ejercicios](#)

Proyectos desarrollados sobre notebooks Jupyter

 Jupyter Notebook ★ 1

[proyectos-laboratorio](#)


Proyectos desarrollados sobre scripts Python

 Python

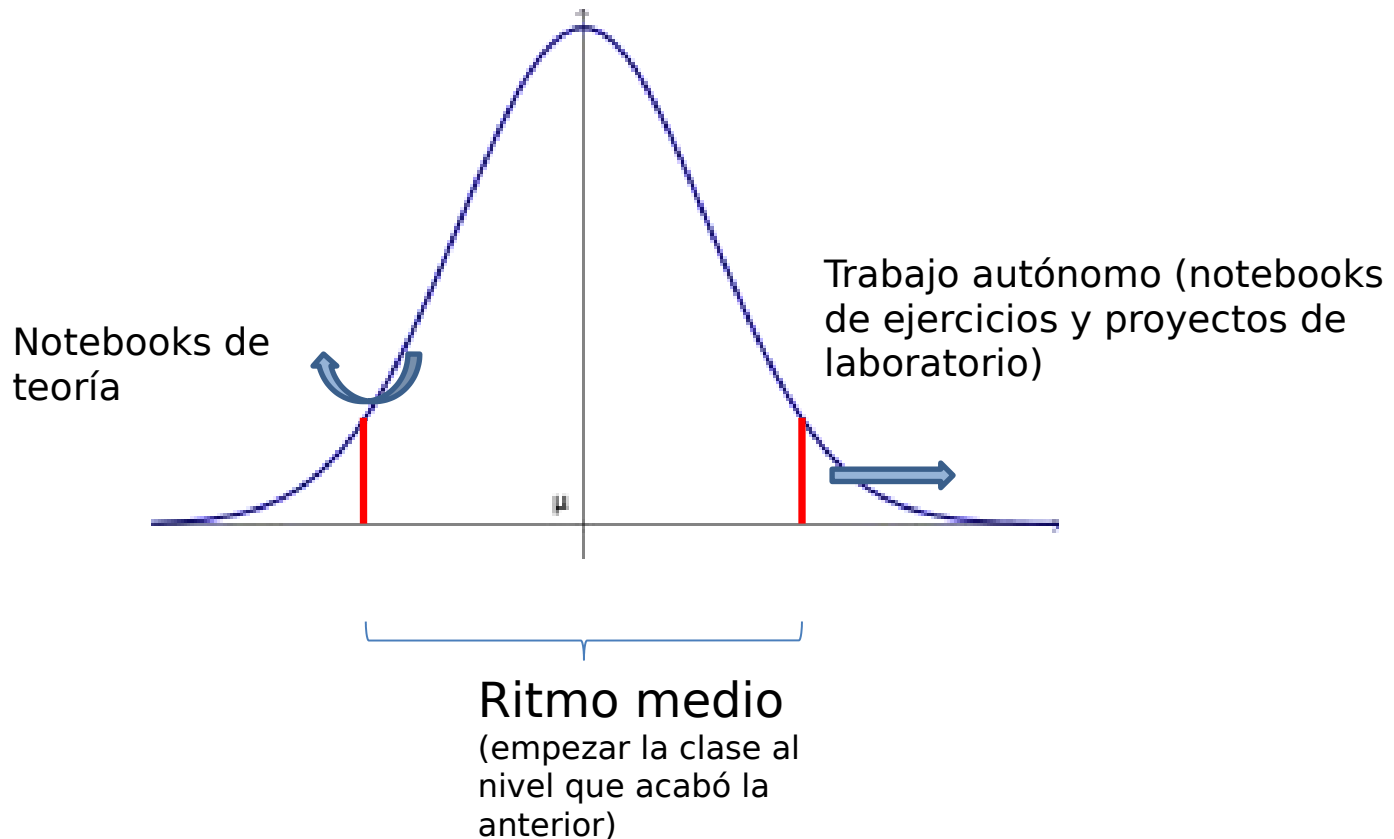
73 contributions in the last year

	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon													
Wed													
Fri													

[Learn how we count contributions.](#)

Less  More

Ritmo de trabajo en clase



No es saber, es saber
buscar



Google



stackoverflow

Y lo más importante ...



... la constancia

Ritmo de trabajo en casa

FP → 12 crs. ECTS

1 cr ECTS → 25 horas de trabajo del alumno

12 crs ECTS → 300 horas de trabajo total

72 horas de teoría y 48 de laboratorio → 130 horas de docencia

¡¡170 horas de trabajo autónomo!!

Asignatura anual → 2 cuatrimestres → 9 meses aprox. → 36 semanas

¡¡4 horas de trabajo autónomo por semana!!

EVALUACIÓN

Evaluación

- Evaluación continua:

$\text{Nota_Cuatrimestre1} = 0,1 \times (C1 + C2 + C3) + 0,6 \times P1 + 0,1 \times \text{PRY1}$ (sobre 10 puntos)

$\text{Nota_Cuatrimestre2} = 0,1 \times (C4 + C5 + C6) + 0,6 \times P2 + 0,1 \times \text{PRY2}$ (sobre 10 puntos)

$\text{Nota_EC} = (\text{Nota_Cuatrimestre1} + \text{Nota_Cuatrimestre2}) / 2$ (sobre 10 puntos)

Ci = Nota cuestionario, Pi = Examen práctico

El estudiante aprueba la asignatura por evaluación continua si se cumplen los tres requisitos siguientes:

- La calificación de cada uno de los dos cuatrimestres es igual o superior a 4 puntos
- La calificación de la evaluación continua es igual o superior a 5 puntos
- Tener APTO en los dos proyectos.



Si **no se supera** la evaluación continua

- Evaluación ordinaria:

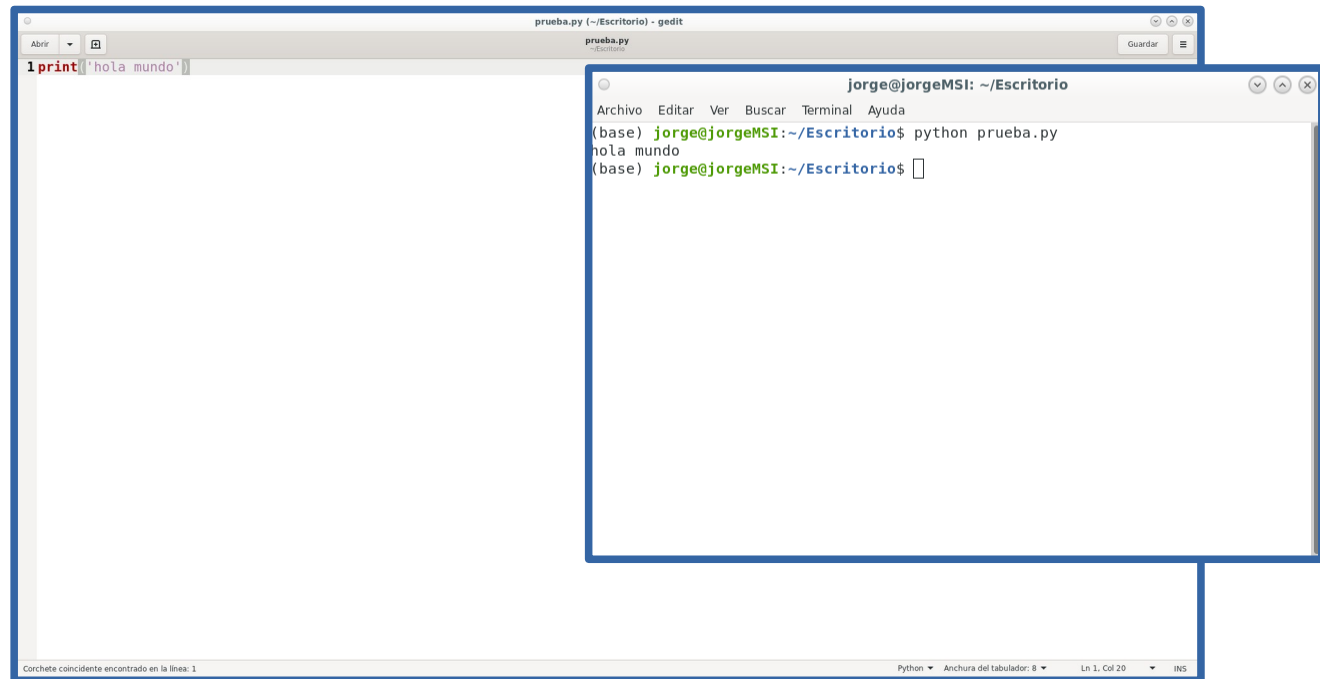
- ✓ Examen final: un **único examen** sobre el ordenador para resolver la implementación de ejercicios prácticos sobre **10 puntos**, que se realizará en el horario establecido por el centro.

Fechas provisionales de las pruebas de evaluación continua

- **Cuestionario 1:** miércoles 5 de octubre
– 12:30 [aula A3.10]
- **Cuestionario 2:** miércoles 9 de noviembre
– 12:30 [aula A3.10]
- **Cuestionario 3:** miércoles 14 de diciembre
– 12:30 [aula A3.10]
- **Prueba de laboratorio:** jueves 19 de enero

TAREA 0

- Descargar documentos de instalación
- Preparar el entorno para poder trabajar en nuestros portátiles o en casa.



The image shows a screenshot of a development environment. In the background, a code editor window titled 'prueba.py (~/Escritorio) - gedit' is open, displaying a single line of Python code: `1 print('hola mundo')`. In the foreground, a terminal window titled 'jorge@jorgeMSI: ~/Escritorio' is open. The terminal shows the command `python prueba.py` being executed, which results in the output `hola mundo`. The terminal window also displays a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. At the bottom of the terminal window, the status bar indicates 'Python', 'Anchura del tabulador: 8', 'Ln 1, Col 20', and 'INS'.

```
prueba.py (~/Escritorio) - gedit
1 print('hola mundo')

jorge@jorgeMSI: ~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
(base) jorge@jorgeMSI:~/Escritorio$ python prueba.py
hola mundo
(base) jorge@jorgeMSI:~/Escritorio$

Corchete coincidente encontrado en la línea: 1
Python Anchura del tabulador: 8 Ln 1, Col 20 INS
```