

Introducción a la programación

Usando Python

G. Sebastián Pedersen

Instituto de Industria
Universidad Nacional de General Sarmiento

Matemática para Economistas III, 1er. cuat. 2019

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.
 - ▶ Frase de Edgar Dijkstra: “La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”.

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.
 - ▶ Frase de Edgar Dijkstra: “La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”.
- ▶ ¿Qué lenguajes de programación conocen?

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.
 - ▶ Frase de Edgar Dijkstra: “La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”.
- ▶ ¿Qué lenguajes de programación conocen?
- ▶ ¿Cuál lenguaje vamos a usar nosotros?

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.
 - ▶ Frase de Edgar Dijkstra: “La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”.
- ▶ ¿Qué lenguajes de programación conocen?
- ▶ ¿Cuál lenguaje vamos a usar nosotros?
 - ▶ Respuesta corta: Python.

Introducción

Info general:

G. Sebastián pedersen – sebasped@gmail.com

Página del curso: <https://sebasped.github.io/pythonungs/>

- ▶ Objetivo: programar un sistema de ecs. difs. y graficarlo.
- ▶ ¿Qué es programar?
 - ▶ Programar \neq saber un lenguaje de programación.
 - ▶ Programar \neq saber usar una computadora.
 - ▶ Frase de Edgar Dijkstra: “La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”.
- ▶ ¿Qué lenguajes de programación conocen?
- ▶ ¿Cuál lenguaje vamos a usar nosotros?
 - ▶ Respuesta corta: Python.
 - ▶ Respuesta larga: no importa demasiado. Lo importante son los conocimientos básicos de programación, que son comunes a la mayoría de los lenguajes.

Recursos Python

► Comunidades:

- <http://www.python.org.ar/>
- <https://argentinaenpython.com/>
- <https://twitter.com/pibesdesistemas>
- <https://www.chicasentecnologia.org/>
- <https://twitter.com/lasdesistemas>
- <https://www.meetup.com/Buenos-Aires-Python-Meetup/>
- <https://twitter.com/linuxchixar>

Recursos Python

► Comunidades:

- <http://www.python.org.ar/>
- <https://argentinaenpython.com/>
- <https://twitter.com/pibesdesistemas>
- <https://www.chicasentecnologia.org/>
- <https://twitter.com/lasdesistemas>
- <https://www.meetup.com/Buenos-Aires-Python-Meetup/>
- <https://twitter.com/linuxchixar>

► Material, cursos, tutoriales, bibliografía:

- <http://www.python.org.ar/wiki/AprendiendoPython>
- <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>
- <https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>
- Cursos online (en inglés): coursera, datacamp, udemy, Stanford online, edx, codeacademy, Harvard online, etc.

Recursos Python

► Comunidades:

- <http://www.python.org.ar/>
- <https://argentinaenpython.com/>
- <https://twitter.com/pibesdesistemas>
- <https://www.chicasentecnologia.org/>
- <https://twitter.com/lasdesistemas>
- <https://www.meetup.com/Buenos-Aires-Python-Meetup/>
- <https://twitter.com/linuxchixar>

► Material, cursos, tutoriales, bibliografía:

- <http://www.python.org.ar/wiki/AprendiendoPython>
- <https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf>
- <https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>
- Cursos online (en inglés): coursera, datacamp, udemy, Stanford online, edx, codecademy, Harvard online, etc.

► Buscar en internet: hay mucho mucho hecho ya.

Python actualmente es muy popular

Índice Tiobe:

<https://www.tiobe.com/tiobe-index/>

TIOBE Index for March 2019

March Headline: Powershell enters the TIOBE index top 50

There are hardly any interesting changes in the TIOBE index this month. We had to use our magnifying glass to spot some news, i.e. that Powershell entered the TIOBE index top 50. The Powershell scripting language is more than 12 years old and it has been in the top 50 before. Powershell is typically used for basic scripting. Until recently it was only available for Windows but Microsoft used its .NET Core platform to create Powershell Core. This version is open source and runs on all major platforms. This might be the reason why Powershell is getting more popular again.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Mar 2019	Mar 2018	Change	Programming Language	Ratings	Change
1	1		Java	14.880%	-0.06%
2	2		C	13.305%	+0.55%
3	4	▲	Python	8.262%	+2.39%
4	3	▼	C++	8.126%	+1.67%
5	6	▲	Visual Basic .NET	6.429%	+2.34%
6	5	▼	C#	3.267%	-1.80%
7	8	▲	JavaScript	2.426%	-1.49%
8	7	▼	PHP	2.420%	-1.59%
9	10	▲	SQL	1.926%	-0.76%
10	14	▲	Objective-C	1.681%	-0.09%

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.
- ▶ ¿Cómo se aprende a programar?

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.
- ▶ ¿Cómo se aprende a programar? Programando... no hay manera de aprender algo sin hacerlo.

Algoritmo y programa

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:

Algoritmo y programa

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.

Algoritmo y programa

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.
- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.

Algoritmo y programa

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.
- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.
 - ▶ El programa representa al algoritmo en el lenguaje.

Algoritmo y programa

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.
- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.
 - ▶ El programa representa al algoritmo en el lenguaje.
 - ▶ Las instrucciones son propias del lenguaje.

```
for i in people.data.users:
    response = client.api.statuses.user_timeline.get(screen_name=i.scre
    print 'Got', len(response.data), 'tweets from', i.screen_name
    if len(response.data) != 0:
        ltdate = response.data[0]['created_at']
        ltdate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S +0000 %Y')
        today = datetime.now()
        howlong = (today-ltdate2).days
        if howlong < daywindow:
            print i.screen_name, 'has tweeted in the past', daywindow,
            totaltweets += len(response.data)
            for j in response.data:
                if j.entities.urls:
                    for k in j.entities.urls:
                        newurl = k['expanded_url']
                        urlset.add((newurl, j.user.screen_name))
        else:
            print i.screen_name, 'has not tweeted in the past', daywindi
```

Proximas diapos

- ▶ Tipos de datos. Operaciones.
- ▶ Asignación.
- ▶ Listas.
- ▶ En cada ítem, iremos mechando algo de práctica
- ▶ Práctica para la clase.