A thick black L-shaped frame surrounds the text. It starts at the top-left, goes right, then down, then right again at the bottom, and finally up on the right side.

# INTRODUCCIÓN A LA PROGRAMACIÓN

## 2.- PYTHON

Rodrigo López A.

rilopez3@uc.cl

# Algoritmo

- Serie de pasos ordenados, bien definidos y finitos para resolver un problema (conseguir algo)



# Descripción de como lavarse los dientes

- Toma tu cepillo de dientes y la pasta dental, verifica que esta ultima se encuentre destapada y procede a untar un porción sobre tu cepillo de dientes, es posible y recomendable humedecer el cepillo con agua. A continuación coloca tu cepillo contra tus dientes formando un ángulo de  $45^{\circ}$  con las encías.
- Mueva el cepillo adelante y atrás suavemente con movimientos cortos de manera repetitiva, de tal manera que las cerdas cubran tus dientes por completo.
- Recuerda cepillar la superficies de tu lengua para eliminar las bacterias y mantener el aliento fresco.
- Verifica que has removido todos los restos de alimentos y procede a enjuagar tu boca

# Algoritmo de como lavarse los dientes (versión x)

- Verificar que se posee 1 cepillo y 1 pasta dental
- Tomar y destapar la pasta dental
- Tomar cepillo de dientes
- Untar una porción de pasta sobre el cepillo
- Tapar y guardar la pasta dental
- Humedecer el cepillo de dientes sobre una fuente de agua disponible, en caso que se prefiera de esta manera
- Introducir el cepillo en nuestras bocas y posicionar contra los dientes en un angulo de 45°
- Cepillar los dientes
- Verificar que se encuentran limpios los dientes
- En caso de no estar limpios aun, volver a cepillar
- Cepillar el resto de la cavidad bucal
- Retirar el cepillo de dientes
- Enjuagar la cavidad bucal
- Enjuagar y guardar el cepillo de dientes

# ¿Cómo pasar desde un Algoritmo a programar?



- **Seudocódigo** es una descripción de alto nivel (en lenguaje natural) compacta e informal del principio operacional de un algoritmo, aplicación o programa.
- Identificar, separar y ordenar los pasos...
- Pasar desde un pseudocódigo a programar es mucho mas sencillo!!

# Seudocódigo de como lavarse los dientes (versión x)

Inicio

1. Verificar que existe 1 cepillo
2. Verificar que existe 1 pasta dental
3. Tomar pasta dental
4. Destapar pasta dental
5. Tomar cepillo de dientes
6. Untar una porción de pasta sobre el cepillo
7. Tapar la pasta dental
8. Guardar pasta dental
9. ¿El usuario prefiera utilizar un cepillo húmedo?  
    *SI: Humedecer el cepillo*  
    *NO: Continuar*
10. Introducir el cepillo en nuestras bocas
11. Posicionar el cepillo contra los dientes

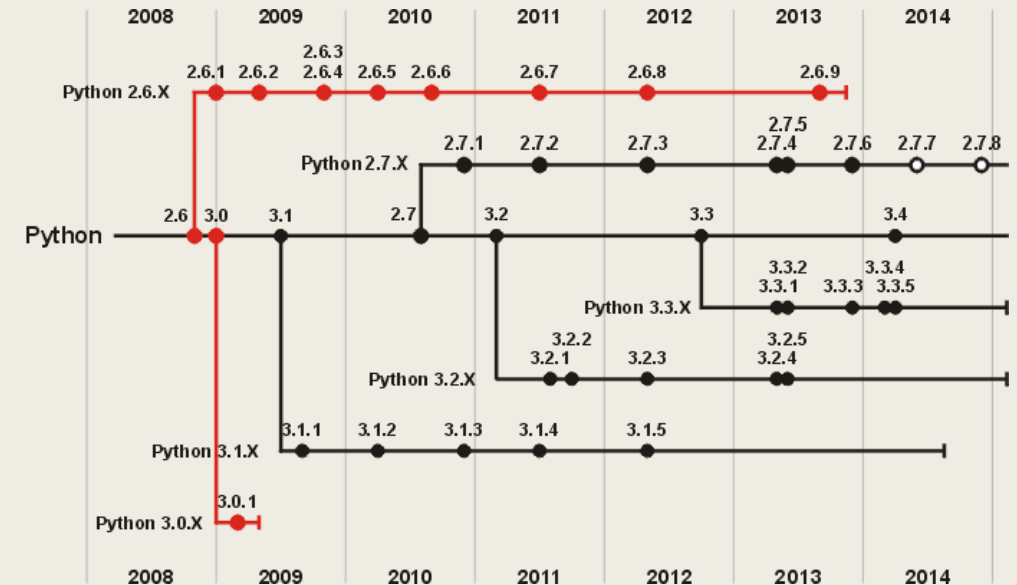
12. ¿Se encuentra el cepillo en un angulo de 45° ?  
    *SI: Continuar*  
    *NO: Volver al paso 11*
13. Mover el cepillo contra los dientes
14. ¿Se han removido todos los restos de alimentos? O ¿Se ha superado un tiempo de 5 minutos durante el cepillado?  
    *SI: Continuar*  
    *NO: Volver al paso 13*
15. Cepillar la superficie de la lengua y el resto de la cavidad bucal
16. Retirar el cepillo de dientes
17. Enjuagar la cavidad bucal
18. Enjuagar el cepillo de dientes
19. Guardar el cepillo de dientes

Termino



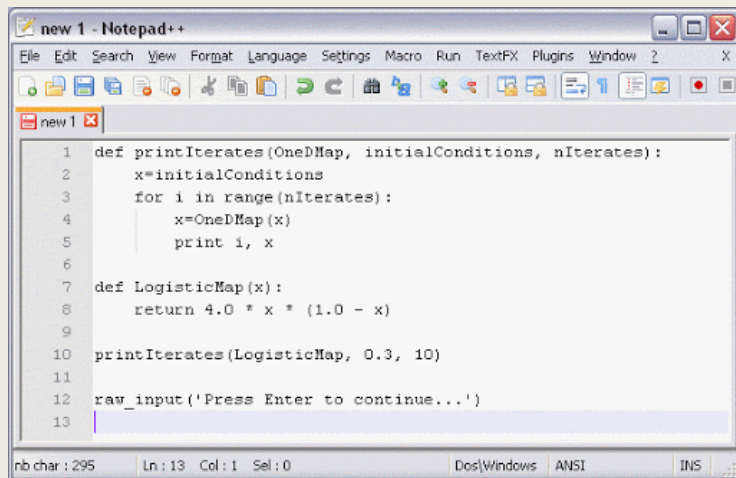
# Python

- Creado a finales de los ochenta (1989) por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos.
- Lanzado al mercado el año 1991
- Es un lenguaje de programación **interpretado**, esto permite que se utilice una semántica muy cercana al lenguaje natural, por lo mismo requiere de un “interprete” para ser utilizado.
- El interprete de Python tiene múltiples versiones...
- Durante el curso usaremos las versiones mas recientes de la versión [3.5 – 3.10]



# Integrated Development Environment (IDE)

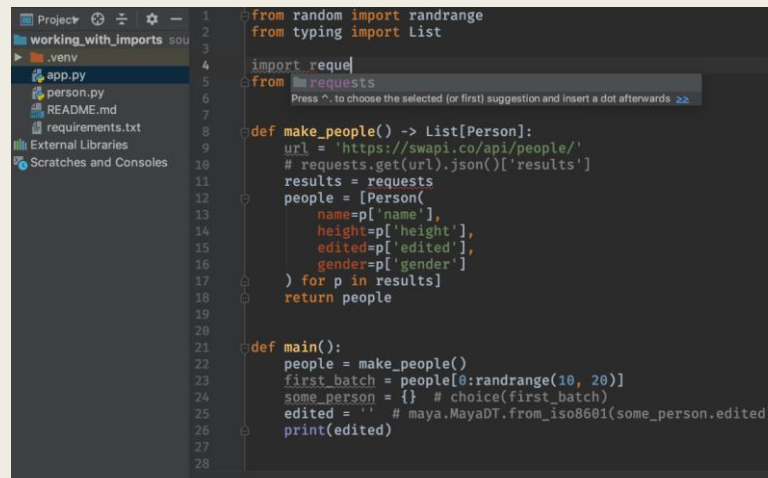
- Programar es redactar un algoritmo en un lenguaje determinado, en nuestro caso en Python
- Para programar podemos utilizar cualquier editor de texto, incluso Notepad (hoy se puede incluso programar desde el celular)
- Existen softwares especializados para programar, estos permiten tener un código ordenado, identificar errores de manera rápida, identificar los componentes, facilitan la integración con otros códigos, lenguajes y aplicaciones
- A estos entornos de programación se les llama IDE, entornos de desarrollo integrado.



A screenshot of the Notepad++ text editor. The window title is 'new 1 - Notepad++'. The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, TextFX, Plugins, Window, and Help. The code is as follows:

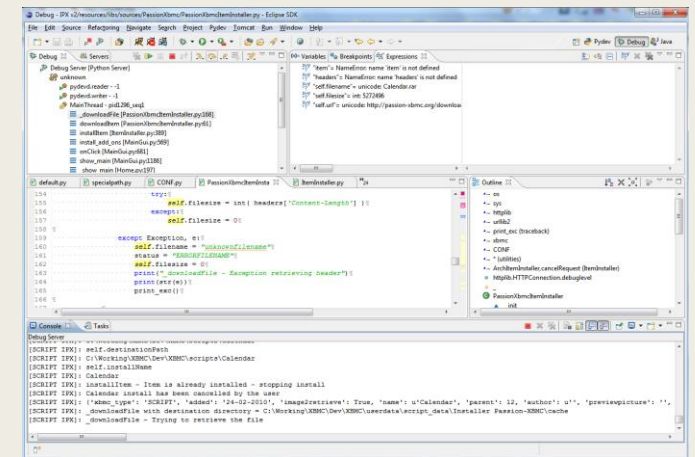
```
1 def printIterates(OneDMap, initialConditions, nIterates):
2     x=initialConditions
3     for i in range(nIterates):
4         x=OneDMap(x)
5         print i, x
6
7 def LogisticMap(x):
8     return 4.0 * x * (1.0 - x)
9
10 printIterates(LogisticMap, 0.3, 10)
11
12 raw_input('Press Enter to continue...')
13
```

The status bar at the bottom shows 'nb char : 295', 'Ln : 13', 'Col : 1', 'Sel : 0', and the file encoding is 'Dos\Windows'.



A screenshot of an IDE, likely PyCharm, showing a Python project. The 'Project' view on the left lists files: app.py, person.py, README.md, requirements.txt, and External Libraries. The main editor shows the following code:

```
1 from random import randrange
2 from typing import List
3
4 import requests
5
6 def make_people() -> List[Person]:
7     url = 'https://swapi.co/api/people/'
8     # requests.get(url).json()['results']
9     results = requests.get(url).json()['results']
10    people = [Person(
11        name=p['name'],
12        height=p['height'],
13        edited=p['edited'],
14        gender=p['gender']
15    ) for p in results]
16    return people
17
18 def main():
19     people = make_people()
20     first_batch = people[:randrange(10, 20)]
21     some_person = {} # choice(first_batch)
22     edited = '' # maya.MayaDT.from_iso8601(some_person.edited)
23     print(edited)
```



A screenshot of an IDE showing a complex Python script. The script is a multi-line program that interacts with a web API and manages a list of people. The console output shows the results of the script execution, including the names and heights of the people retrieved from the API.

```
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



# Tipos de IDE y elección

- Existen múltiples tipos de IDE:
  - *diferentes niveles de complejidad*
  - *diferentes UI (User Interface)*
  - *locales o web*
  - *con colores, con destacados, sin colores...*
  - *con detección de errores o depuración (revisión paso a paso)*
  - *con distintas integraciones*
- ¿Cómo escoger? o ¿Cuál es mejor?
  - *No existen IDEs mejores o perfectos, todo depende de cual te facilita las tareas que requieres*
- Durante el curso se recomienda el uso de 2 IDEs
  - *Local: PyCharm (sencillo y fácil de usar, versión gratuita para estudiantes)*
  - *Web: Google Colab (entorno web)*

# ¿Qué se requiere para programar en Python?

## ■ Local:

- *Al menos 1 interprete*
- *Se puede utilizar los editores de texto normales o un IDE*
- *Puede requerir configuraciones en muchos casos...*

*¿Como se instala? Les enviaremos una guía paso a paso, con algunas funcionalidades útiles*



## ■ Web:

- *Un navegador web*

*¿Cómo acceder? Se requiere solo una cuenta*

*Google y acceder al sitio <https://colab.research.google.com/>*



# Tipos de programación con Python

- Programación en consola (horrible...):
  - *Desde la consola de Windows (CMD) o el prompt de Linux o Mac, se introducen las instrucciones*
- Ejecución de archivos (método popular):
  - *Se redactan en un archivo todas las instrucciones, y se ejecuta el archivo por completo*
- Uso de cuadernos Jupyter notebook (perfecto para aprender):
  - *Es posible ejecutar y probar paso a paso cada instrucciones*