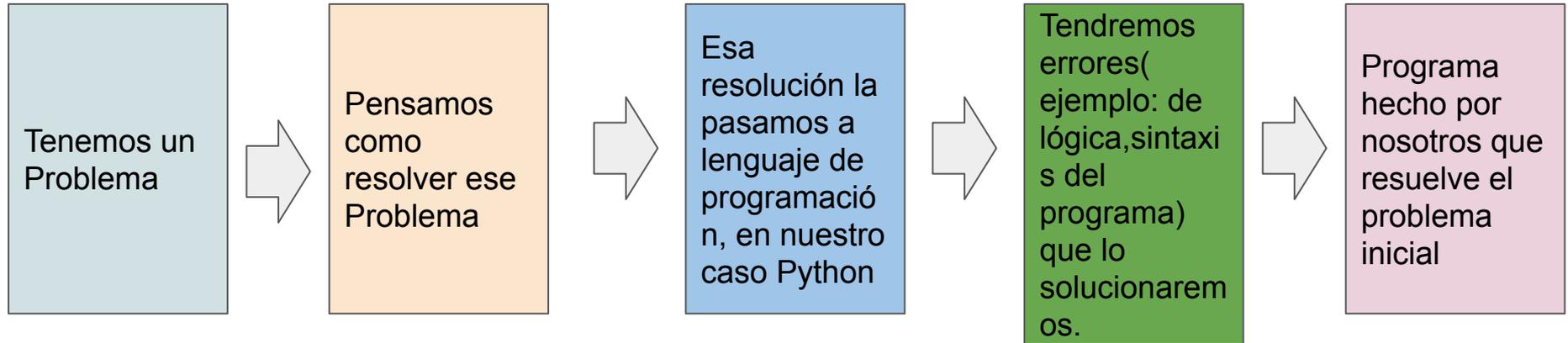


EL proceso simplificado para programar



Lección 3.1: Definiciones más Importantes

- **Variables**: es un lugar donde guardamos una determinada información. En función del tipo de información **que** guardemos (texto, números, booleanas, etc.).

- **Tipos de Dato**: Es un atributo de los **datos** que indica al ordenador (y/o al programador) sobre la clase de **datos** que se va a trabajar.

- **Código Fuente**: Es el código que el programador escribirá en un archivo de acuerdo al lenguaje de programación en nuestro caso python, por lo tanto el archivo tendrá extensión .py

- **Lenguaje de Tipado Dinámico**: Es cuando la comprobación de tipificación se realiza durante su ejecución en vez de durante la compilación

Lección 3.1: Definiciones más Importantes

-**algoritmo informático** :es una secuencia de instrucciones finitas que llevan a cabo una serie de procesos para dar respuesta a determinados problemas.

-Lenguaje de alto Nivel: Es el lenguaje que entiende el programador, en este caso, nosotros aprenderemos a programar en el lenguaje de programación Python.

-Lenguaje Máquina: es el lenguaje que entiende la computadora(específicamente el procesador) , es decir, el código binario(0 y 1)

Lenguaje de Alto Nivel

```
▶ diccionario={  
    'juan':20,  
    'martin':34,  
    'julian':10  
}  
diccionario
```

```
{'juan': 20, 'julian': 10, 'martin': 34}
```

```
▶ for ronda in range(1,3):  
    if(ronda==1):  
        diccionario['juan']=diccionario['juan'] + 12  
        diccionario['martin']=diccionario['martin']+ 12  
        diccionario['julian']=diccionario['julian'] + 12  
    else:  
        diccionario['juan']=diccionario['juan'] + 30  
        diccionario['martin']=diccionario['martin']+ 30  
        diccionario['julian']=diccionario['julian'] + 30  
  
print(diccionario)  
#aca estamos repitiendo codigo, lo veremos como solucionarlo  
#cuando veamos funciones
```

```
↳ {'juan': 62, 'martin': 76, 'julian': 52}
```

Lenguaje Máquina

```
[0x00000000]> pd
0x00000000  90          nop
0x00000001  90          nop
0x00000002  6800009c00 push 0x9c0000 ; 0x009c0000
0x00000007  e8c7ace37b call 0x7be3acd3
                0x7be3acd3(unk)
0x0000000c  bb04009c00 mov ebx, 0x9c0004
0x00000011  8903       mov [ebx], eax
0x00000013  e81903f47b call 0x7bf40331
                0x7bf40331( )
0x00000018  bb08009c00 mov ebx, 0x9c0008
0x0000001d  8903       mov [ebx], eax
0x0000001f  bb00009c00 mov ebx, 0x9c0000
0x00000024  c60300    mov byte [ebx], 0x0
-> 0x00000027  68e8030000 push 0x3e8 ; 0x000003e8
| 0x0000002c  e81124e37b call 0x7be32442
|                0x7be32442(unk)
=< 0x00000031  ebf4       jmp 0x10000027
0x00000033  90          nop
0x00000034  ff         invalid
0x00000035  ff         invalid
0x00000036  ff         invalid
0x00000037  ff         invalid
```

Python: Lenguaje Interpretado

-Python es un lenguaje Interpretado: Es decir , es convertido a lenguaje máquina a medida que es ejecutado el código fuente. Pensarlo como un escáner que está escaneando una foto, mientras va pasando su láser, va formando su parte de la imagen mientras se ejecuta.

```
12 pesos, y la segunda ronda 30.
1)Mostrar el total acumulado de cada cliente despues de la segunda

diccionario={
    'juan':20,
    'martin':34,
    'julian':10
}
diccionario
{'juan': 20, 'julian': 10, 'martin': 34}

for ronda in range(1,3):
    if(ronda==1):
        diccionario['juan']=diccionario['juan'] + 12
        diccionario['martin']=diccionario['martin']+ 12
        diccionario['julian']=diccionario['julian'] + 12
    else:
        diccionario['juan']=diccionario['juan'] + 30
        diccionario['martin']=diccionario['martin']+ 30
        diccionario['julian']=diccionario['julian'] + 30

print(diccionario)
#aca estamos repitiendo codigo, lo veremos como solucionarlo
#cuando veamos funciones

{'juan': 62, 'martin': 76, 'julian': 52}
```



```
[0x00000000]> pd
0x00000000  90          nop
0x00000001  90          nop
0x00000002  6800009c00 push 0x9c0000 ; 0x009c0000
0x00000007  e8c7ace37b call 0x7be3acd3
0x7be3acd3(unk)
0x0000000c  bb04009c00 mov ebx, 0x9c0004
0x00000011  8903       mov [ebx], eax
0x00000013  e81903f47b call 0x7bf40331
0x7bf40331(unk)
0x00000018  bb08009c00 mov ebx, 0x9c0008
0x0000001d  8903       mov [ebx], eax
0x0000001f  bb00009c00 mov ebx, 0x9c0000
0x00000024  c60300    mov byte [ebx], 0x0
-> 0x00000027  68e8030000 push 0x3e8 ; 0x000003e8
0x0000002c  e81124e37b call 0x7be32442
0x7be32442(unk)
=< 0x00000031  eb44       jmp 0x100000027
0x00000033  90          nop
0x00000034  ff        invalid
0x00000035  ff        invalid
0x00000036  ff        invalid
0x00000037  ff        invalid
```

Lenguaje Compilado(estos es información adicional)

Tienen la particularidad que requiere un paso adicional antes de ser ejecutado, la compilación, que convierte el código que escribes a lenguaje de máquina. Una vez convertido a lenguaje máquina, este recién puede ser ejecutado para mostrar un resultado, ect. Ejemplos: c