

Auxiliar 1 - Autómatas finitos y no deterministas

Profesor: Alejandro Hevia
Auxiliares: Nicolás Lehmann [nlehmann@dcc.uchile.cl]
José Garrido

P1. La expansión en serie de Taylor de la función $\cos(x)$ en torno al cero es:

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

Considerando hasta el cuarto término podemos aproximar la función de la siguiente forma:

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}$$

Escriba una expresión en python que calcule aproximadamente el valor $\cos(0.785)$. ¿Qué harías si tienes que calcular la función en más de un punto del dominio?

P2. Considere los programas listados a continuación. ¿Qué imprime el interprete en cada caso?

```
>>> a = 1
>>> b = 2
>>> a b +

>>> a = "Hola"
>>> a * "tarola"

>>> def sumToN(n):
...     a = n*(n+1)
...     return a/2

>>> def foo(x):
...     return x/2 - 1
>>> a = foo(3)
>>> 8/a
```

P3. Utilizando la “Receta de Diseño” vista en clases. Defina la función **pesosADolar** que recibe un valor numérico correspondiente a la cantidad que se tiene en Pesos y que se desean transformar en Dolares. Considere que: 1 Dolar = 571 Pesos

Modifique la función **pesosADolar** de manera de que dicha función reciba 2 argumentos, el primero de ellos corresponde a la cantidad en Pesos que se desea transformar, mientras que el segundo argumento corresponde al factor de cambio que se utilizará.

¿Cómo podría generar una función que generalice el cambio de una divisa por otra entregando la cantidad y el factor de cambio?

P4. Considere los programas listados a continuación. ¿Qué imprime el interprete en cada caso? Explique la razón del comportamiento obtenido en cada caso.

```
>>> densidad_cobre = 8960
>>> def volumen_cobre(masa):
...     return masa/densidad_cobre
>>> volumen_cobre(2500)
>>> densidad_cobre = 8960.0
>>> volumen_cobre(2500)
```

```
>>> g = 9.78
>>> def peso(masa):
...     g = 10
...     return masa*g
>>> peso(50)
>>> g
```

```
>>> def presion(fuerza):
...     area = 100
...     return fuerza/area
>>> presion(1000)
>>> area
```