

□ Cuestión 2: Las funciones y el comando block

☒ Esta función calcula el valor de f en el punto medio de [a,b]

```
☒ --> kill(all);
```

```
☒ --> prueba(fn,a,b):=block([c],  
    c:(a+b)/2,  
    fn(c)  
    )$
```

☒ Probamos con la función cos(x)

```
☒ --> f(x):=cos(x)$
```

```
☒ --> prueba(f,0,%pi);
```

☒ Parece que lo hace bien, pero definamos una función que se llame fn

```
☒ --> fn(x):=1$
```

☒ Y ejecutamos la misma instrucción de antes

```
☒ --> prueba(f,0,%pi);
```

☒ Resulta que "prueba", en lugar de considerar el parámetro "fn" que le hemos pasado, ha tomado la función global que hemos definido. Veamos como resolver esta "pega".

```
☒ --> prueba(fn,a,b):=block([c],  
    local(fn),  
    c:(a+b)/2,  
    fn(c)  
    )$
```

```
--> prueba(f,0,%pi);
```

Podría resultar válido introducir "fn" en el corchete junto a "c". Sin embargo, esto conduciría de nuevo a la respuesta errónea.

```
--> prueba(fn,a,b):=block([c,fn],
    c:(a+b)/2,
    fn(c)
)$
```

```
--> prueba(f,0,%pi);
```

Sí parece adecuado introducir "c" en "local". Sin embargo, también esto conduciría a una situación no deseada: "c" olvidaría el valor previo si lo hubiera.

```
--> c:1;
```

```
--> prueba(fn,a,b):=block(
    local(c,fn),
    c:(a+b)/2,
    fn(c)
)$
```

```
--> prueba(f,0,%pi);
```

```
--> c;
```

Aunque no se vería afectada "fn".

```
--> fn(2);
```