Paquetes y módulos

Los módulos son archivos de programas que contienen código útil. Los paquetes son conjuntos de módulas con funcionalidades similares. Las librerías son conjuntos de paquetes.

En Julia los módulos requieren de exportar ciertos nombres para poder usarlos dentro de otro ambiente.

Namespace

Un *namespace* es un conjunto de nombres reservados en un programa. Así, por ejemplo, los nombres reservados en un módulo difieren de los nombres de otro módulo. Al trabajar en REPL se considera que estamos en el *global namespace*.

> 💡Para acceder a nombres de objetos que pertenecen a otro namespace del actual, utilizamos un *qualified name*, por ejemplo si `a` se encuentra en el namespace `Something`, el nombre es `Something.a`.

Módulos

Para utilizar módulos *previamente instalados* en Julia, podemos utilizar las siguientes opciones.

`using`

La sentencia `using` pone en disponibilidad todos los nombres exportados del módulo en el namespace actual.

```
using Module1, ...
```

No hay necesidad de utilizar nombres calificados ya que se encuentran dentro de nuestro namesapce, sin embargo, Julia lanzará error si creamos un nuevo objeto con un nombre ya utilizado.

Los nombres que no se han exportado en el módulo aún pueden utilizarse mediante un nombre calificado.

```
Module.object
```

`import`

La sentencia `import` pone en disponibilidad todos los nombres, pero siempre requiere de un nombre calificado.

```
import Module1, ...
```

Renombrar

se puede renombrar un módulo con el keyword `as`. Por ejemplo:

```
import Module as NewName
```

Todas las referencias al módulo requerirán de `NewName`. También es posible renombrar algún objeto del módulo:

```
import Module: object as NewName
```

Creando módulos

Un módulo puede definirse mediante varios archivos y un archivo puede contener varios módulos. Un módulo se encuentra en un bloque:

```
module NameModule
    code
end
```

Tenemos las siguientes consideraciones:

- Las indentaciones se comprimen en un archivo.
- Cada definición dentro del módulo, y exterior a otros bloques, se considera global al módulo.


En Julia *siempre* estamos en un módulo. El módulo a nivel más alto se denomina `Main`.

Además, la notación con puntos empieza a ser similar a los sistemas UNIX. Por ejemplo, sea M1 un módulo en el *Main Module*:

```
import .M1
```

La sentencia indica que se importa el módulo `M1` a nivel de `Main`. Siguiendo, sea `M2` un módulo definido dentro de `M1`:

```
module M2
    ...
    import ..M1
    ...
end
```

La sentencia indica que se importa `M1`, que se encuentra un nivel superior al nivel de `M2`. Para importar `M2` en el nivel `Main`:

```
import .M1.M2
```

Importar `M2`, que pertenece al namespace de `M1` en el nivel `Main`.

Documentando con Docstrings

Al crear un módulo se exportan los nombres deseados preferentemente en la línea más superior del módulo.

```
module Name
    export object1, ...
    ...
```

Se puede documentar un módulo con el sistema Docstring, que comprende una version de Markdown. Para documentar, se utilizan tres comillas dobles antes de la definición de una función:

```
module Name
export ...

"""
Markdown
"""

function f()
    ...
end

end
```