Entrée [1]:

```
println("Hello World")
```

```
Hello World
```

Entrée [10]:

```
123
456
```

Out[10]:

```
456
```

Entrée [12]:

```
my_answer = 42
typeof(my_answer)
```

Out[12]:

```
Int32
```

Entrée [4]:

```
my_pi = 3.14159
typeof(my_pi)
```

Out[4]:

```
Float64
```

Entrée [5]:

```
my_name = "Ahmet"
typeof(my_name)
```

Out[5]:

```
String
```

Entrée [7]:

```
my_answer = my_name
```

Out[7]:

```
"Ahmet"
```

Entrée [8]:

```
typeof(my_answer)
```

Out[8]:

```
String
```

Entrée [13]:

```
# la syntaxe du commentaire d'une ligne
```

Entrée [14]:

```
#=

La syntaxe pour plisieur
ligne de commentaire

=#
```

Entrée [15]:

```
3+2
```

Out[15]:

```
5
```

Entrée [16]:

```
10-5
```

Out[16]:

```
5
```

Entrée [22]:

```
20*5
```

Out[22]:

```
100
```

Entrée [18]:

```
100/10
```

Out[18]:

```
10.0
```

Entrée [19]:

```
10^2
```

Out[19]:

```
100
```

Entrée [21]:

```
101%2
```

Out[21]:

```
1
```

Entrée [26]:

```
"""Test, voir "error"!!!"""
```

Out[26]:

```
"Test, voir \"error\"!!!"
```

Entrée [33]:

```
"""Test "ahaaha","voir" ou "test"!!"""
```

Out[33]:

```
"Test \"ahaaha\",\"voir\" ou \"test\"!!"
```

Entrée [35]:

```
"L'erreur "et"!!"
```

```
syntax: cannot juxtapose string literal

Stacktrace:
 [1] top-level scope
   @ In[35]:1
 [2] eval
   @ .\boot.jl:360 [inlined]
 [3] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::Stri
ng, filename::String)
   @ Base .\loading.jl:1094
```

Entrée [37]:

```
name = "Ahmet"
nombre_doigt = 10
nombre_ongle = 10
println("Bonjour, je m'appelle is $name.")
println("J'ai $nombre_doigt et $nombre_ongle. Donc le total est $(nombre_doigt + nombre_ong
```

```
Bonjour, je m'appelle is Ahmet.
J'ai 10 et 10. Donc le total est 20
```

Entrée [44]:

```
s1 = "Comment tu t'appele ?";
s2 = "je m'appele ahmet";
```

Entrée [43]:

```
s1*s2
```

Out[43]:

```
"Comment tu t'appele ?je m'appele ahmet"
```

Entrée [41]:

```
"$s1$s2"
```

Out[41]:

```
"Comment tu t'appele ?je m'appele ahmet"
```

Entrée [46]:

```
montelephone = Dict("Ahmet"=>"412-4120", "Test1" =>"512-5120")
```

Out[46]:

```
Dict{String, String} with 2 entries:
  "Test1" => "512-5120"
  "Ahmet" => "412-4120"
```

Entrée [47]:

```
montelephone["Joseph"] = "111-test"
```

Out[47]:

```
"111-test"
```

Entrée [48]:

```
montelephone
```

Out[48]:

```
Dict{String, String} with 3 entries:
  "Test1"  => "512-5120"
  "Joseph" => "111-test"
  "Ahmet"  => "412-4120"
```

Entrée [49]:

```
pop!(montelephone,"Test1")
```

Out[49]:

```
"512-5120"
```

Entrée [50]:

```
montelephone
```

Out[50]:

```
Dict{String, String} with 2 entries:
  "Joseph" => "111-test"
  "Ahmet"  => "412-4120"
```

Entrée [51]:

```
montelephone[1]
```

```
KeyError: key 1 not found

Stacktrace:
 [1] getindex(h::Dict{String, String}, key::Int32)
   @ Base .\dict.jl:482
 [2] top-level scope
   @ In[51]:1
 [3] eval
   @ .\boot.jl:360 [inlined]
 [4] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::Stri
ng, filename::String)
   @ Base .\loading.jl:1094
```

Entrée [52]:

```
animaux = ("lyon","chat","panthere")
```

Out[52]:

```
("lyon", "chat", "panthere")
```

Entrée [53]:

```
animaux[1]
```

Out[53]:

```
"lyon"
```

Entrée [ ]:

```

```