Единственное, что здесь действительно стоит заметить, так это то, что код представляет из себя набор вложенных S-выражений: (функция аргумент1 аргумент2 ...). Запятые ничем не отличаются от пробелов. Код также есть в файле basics.clj. Точки с запятой начинают отнострочные комментарии.

```
123
=>  123

1.0
=>  1.0

(+ 1 2 3)
=>  6

(+)
=>  0

(+ 1, 2, 3)
=>  6

(- 5 2)
=>  3

(/ 3 2)
=>  3/2

(/ 6 4)
=>  3/2

1.5
=>  1.5

(= 1.5 3/2)
=>  false

(/ 3 2)
=>  3/2
```

```
3/2
```

```
=>  3/2
```

```
(* 3/2 2)
```

```
=>  3N
```

```
3
```

```
=>  3
```

```
3N
```

```
=>  3N
```

```
(type 3/2)
```

```
=>  clojure.lang.Ratio
```

```
(+ 8000000000000000000 8000000000000000000)
```

```
Execution error (ArithmeticException) at java.lang.Math/addExact (Math.java:931) .
long overflow
```

```
(+' 8000000000000000000 8000000000000000000)
```

```
=>  16000000000000000000N
```

```
(type 3)
```

```
=>  java.lang.Long
```

```
(type 3N)
```

```
=>  clojure.lang.BigInt
```

```
(/ 1 0)
```

```
Execution error (ArithmeticException) at user/eval2017 (form-
init17984610524906141998.clj:1) .
Divide by zero
```

```
(/ 1.0 0.0)
```

```
=>  ##Inf
```

```
(+ 0.1 0.2)
```

```
=>  0.30000000000000004
```

```
"abc"
=>  "abc"

(type "abc")
=>  java.lang.String

\a
=>  \a

(type \a)
=>  java.lang.Character

\tab
=>  \tab

(list 1 2 3)
=>  (1 2 3)

 (1 2 3)
Execution error (ClassCastException) at user/eval2047 (form-
init17984610524906141998.clj:1) .
class java.lang.Long cannot be cast to class clojure.lang.IFn (java.lang.Long is in
module java.base of loader 'bootstrap' ; clojure.lang.IFn is in unnamed module of
loader 'app')

(* (+ 2 3) 4)
=>  20

(str 1)
=>  "1"

(str "abc" "def")
=>  "abcdef"

(str (str 1) (str 2))
=>  "12"

(str 1 2)
=>  "12"
```

```
(type (list 1 2 3))
=>  clojure.lang.PersistentList

[1 2 3]
=>  [1 2 3]

(type [1 2 3])
=>  clojure.lang.PersistentVector

(def x [1 2 3])
=>  #'user/x

x
=>  [1 2 3]

user/x
=>  [1 2 3]

(nth x 1)
=>  2

(cons 4 x)
=>  (4 1 2 3)

(vec (cons 4 x))
=>  [4 1 2 3]

(conj 4 x)
Execution error (ClassCastException) at user/eval2131 (form-
init17984610524906141998.clj:1) .
class java.lang.Long cannot be cast to class clojure.lang.IPersistentCollection
(java.lang.Long is in module java.base of loader 'bootstrap' ;
clojure.lang.IPersistentCollection is in unnamed module of loader 'app')


(conj x 4)
=>  [1 2 3 4]

(conj (list 1 2 3) 4)
=>  (4 1 2 3)
```

```
(assoc x 1 4)
=>  [1 4 3]

x
=>  [1 2 3]

(def x 2)
=>  #'user/x

x
=>  2

true
=>  true

false?
=>  #object[clojure.core$false_QMARK_ 0x192e860b
"clojure.core$false_QMARK_@192e860b"]

false
=>  false

(not true)
=>  false

(and true false)
=>  false

(or true false)
=>  true

(if true 23 45)
=>  23
```

```
(def fact
    (fn [n]
        (if
            (< n 1)
            1
            (* n (fact (- n 1))))))
```

```
=>  #'user/fact
```

```
(fact 5)
```

```
=>  120
```

```
(def fact
    (fn [n]
        (if
            (< n 1)
            1
            (* n (recur (- n 1))))))
```

```
=> Syntax error (UnsupportedOperationException) compiling recur at (/tmp/form-
init17984610524906141998.clj:6:12) .
Can only recur from tail position
```

```
(def fact
    (fn [n m]
        ))
```

```
=>  #'user/fact
```

```
(print 1)
```

```
1
```

```
=> nil
```

```
(def fact
    (fn [n m]
        (if
            (< n 1)
            m
            (recur (- n 1) (* n m)))))
```

```
=>  #'user/fact
```

```
(fact 5 1)
```

```
=>  120
```

```
(fact 5 2)
```

```
=>  240
```

```
(def factorial (fn [n] (fact n 1)))
```

```
=>  #'user/factorial
```

```
(factorial 6)
```

```
=>  720
```

```
(defn factorial [n] (fact n 1))
```

```
=>  #'user/factorial
```

```
(filter
    (fn [x] (= 1 (rem x 2)))
    [1 2 3 4 5 6 7 8])
```

```
=>  (1 3 5 7)
```

```
(filter
    #(= 1 (rem % 2))
    [1 2 3 4 5 6 7 8])
```

```
=>  (1 3 5 7)
```

```
(filter
    odd?
    [1 2 3 4 5 6 7 8])
```

```
=>  (1 3 5 7)
```

```
(filter
    even?
    [1 2 3 4 5 6 7 8])
```

```
=>  (2 4 6 8)
```

```
(map
    #(* % %)
    (range 8))
```

```
=>  (0 1 4 9 16 25 36 49)
```

```
(range)
```

И-и-и... REPL зависла. `range` на самом деле возвращает бесконечную последовательность, которую REPL пытается преобразовать в строку, чтобы напечатать.

```
(take-while #(< % 128) (map #(* % %) (range)))
```

```
=> (0 1 4 9 16 25 36 49 64 81 100 121)
```

```
(type (take-while #(< % 128) (map #(* % %) (range))))
=>  clojure.lang.LazySeq

(map #(print %) (range 100))
0123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354
=> (nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil
nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil
nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil
nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil
nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil nil)

(take 1 (map #(print %) (range 100)))
0123456789101112131415161718192021222324252627282930 31
=> (nil)
```

Откуда здесь 32 числа? `map` действует лениво, но не очень. Она бьёт последовательность на блоки по 32 элемента и преобразует их по требованию, блоками.

```
(defn square [x] (* x x))
=>  #'user/square

(defn trice [x] (* x 3))
=>  #'user/trice

(comp square trice)
=> #object[clojure.core$comp$fn__5888 0x17af57d7
"clojure.core$comp$fn__5888@17af57d7"]

(def very-special-func (comp square trice))
=>  #'user/very-special-func

(very-special-func 4)
=>  144

((comp square trice) 4)
=>  144

(defn cube [x] (* x x x))
=>  #'user/cube
```

```
(-)
```

```
Execution error (ArityException) at user/eval1974 (form-init10262687158678453236.clj:
1) .
Wrong number of args (0) passed to: clojure.core/-
```

```
(- 2)
```

```
=>  -2
```

```
(- 5 3)
```

```
=>  2
```

```
(- 5 3 2)
```

```
=>  0
```

```
((#(comp % %) #(* % %)) 3)
```

```
=>  81
```

```
((comp) 3)
```

```
=>  3
```

```
(identity 3)
```

```
=>  3
```

```
((constantly 5))
```

```
=>  5
```

```
(constantly 5)
```

```
=>  #object[clojure.core$constantly$fn__5752 0x78ac6be6
"clojure.core$constantly$fn__5752@78ac6be6"]
```

```
(def const5 (constantly 5))
```

```
=>  #'user/const5
```

```
(const5)
```

```
=>  5
```

```
(const5 1 2 3 4 6 7)
```

```
=>  5
```

```
(+ 1 2 3)
=>  6

(def l (list 1 2 3))
=>  #'user/l

(apply + l)
=>  6

(partial - 1)
=>  #object[clojure.core$partial$fn__5920 0x26b546ec
"clojure.core$partial$fn__5920@26b546ec"]

(def func (partial - 1))
=>  #'user/func

(func 3)
=>  -2

(map #(+ 2) [1 2 3])
=> Error printing return value (ArityException) at clojure.lang.AFn/throwArity
(AFn.java:429) .
Wrong number of args (1) passed to: user / eval2046/fn--2047

(map #(+ 2 %) [1 2 3])
=>  (3 4 5)

(map (partial + 2) [1 2 3])
=>  (3 4 5)

(map list [1 2 3] [4 5 6])
=>  ((1 4) (2 5) (3 6))
```