

# Условные конструкции

**if**                      «если – то – иначе»

**cond**                    «по-условию»

**when**                   «если-то» ( «когда» )

**unless**                «если-не – то»

**case**                   «если-значение-равно...»

# if

## Синтаксис

```
if test-form  
    then-form  
    [else-form]  
=> result
```

## Пример

```
(if (atom x) "x is atom"  
    "x is not atom")
```

# If (пример функции)

```
(defun sum_or_product (a b)
  (if (= a b) (+ a b) (* a b))
)
```

```
(defun sum_or_product1 (a b)
  (if (= a b) (+ a b))
)
```

```
(sum_or_product 3 3) => 6
```

```
(sum_or_product1 3 4) => nil
```

# cond

## Синтаксис

```
cond {clause}* => result*
```

```
clause ::= (test-form then-form*)
```

## Пример

```
(cond  
  ((= x 1) "x is 1")  
  ((atom x) "x is atom")  
  (t "x is very complex thing")  
)
```

# cond (пример)

```
(defun cond_example(a b)
  (cond
    ((or (not (number a))
          (not (number b))) "error")
    (= a b) "a is equal b")
    (< a b) "a less b")
    (t "b less a")
  )
)
```

# when

## Синтаксис

`when test-form form => result`

## Пример

```
(when (atom x) "x is atom")
```

# unless

## Синтаксис

`unless test-form form => result`

## Пример

```
(unless (atom x) "x is atom")
```

# case

## Синтаксис

```
case keyform
  {normal-clause}*
  [otherwise-clause] => result
```

## Пример

```
(case x
  ((+ - / *) "арифметическая функция")
  ((not or and) "логическая функция")
  (log "логарифмическая функция")
  ((sin cos tan) "тригонометрическая функция")
  ((t) "логическая константа истина")
  (t "нераспознанная функция")
)
```