

# Podmínky a cykly v R

## if()

- Rozhodovací ovládací prvek založený na pravdivosti, či nepravdivosti nějakého výroku.
- Else() je nepovinné

```
if(x == 1){  
  print("x je rovno 1")  
} else {  
  print("x není rovno 1")  
}
```

## for()

- Smyčka pro opakování procedury stejného charakteru
- Vhodná tehdy, víme-li dopředu počet opakování (iterací) dané procedury

```
# vytiskne postupně čísla od 1 do 5  
for(i in 1:5){  
  print(i)  
}  
  
# letters obsahuje písmena a-z  
for(my_letter in letters){  
  print( paste(my_letter, "je fajn", sep = " ") )  
}
```

## while()

- Smyčka pro opakování procedury stejného charakteru
- Vhodná tehdy, nevíme-li dopředu počet opakování (iterací) dané procedury

```
i <- 1  
while(i <= 5){  
  print(i)  
  i <- i + 1  
}  
  
# anebo  
my_letters <- letters  
while(length(my_letters) > 0){  
  print( paste(my_letters[1], "je fajn", sep = " ") )  
  my_letters <- my_letters[-1]  
}
```

## repeat-until cyklus

- Smyčka pro opakování procedury stejného charakteru
- Vhodná tehdy, nevíme-li dopředu počet opakování (iterací) dané procedury
- První cyklus se provede vždy
- Jde o smyčku while() použitou jiným způsobem

```
i <- 1  
while(TRUE){  
  if(i == 5){break}  
  print(i)  
  i <- i + 1  
}  
  
# anebo  
my_letters <- letters  
while(TRUE){  
  if(length(my_letters) == 0){break}  
  print( paste(my_letters[1], "je fajn", sep = " ") )  
  my_letters <- my_letters[-1]  
}
```

## Náhrada for cyklu funkcí lapply()

Obě procedury jsou ekvivalentní stran výstupu, lapply() je ale významně rychlejší.

```
# for cyklus  
my_start <- Sys.time()  
for_x <- NULL  
for(i in 1:100000){for_x <- c(for_x, i ^ 5)}  
my_stop <- Sys.time(); my_stop - my_start # 18.45s
```

```
# lapply
my_start <- Sys.time()
lapply_x <- unlist(lapply(
  1:100000, function(i) i ^ 5
))
my_stop <- Sys.time(); my_stop - my_start # 0.10s
```

## Odkazy

### Související články

- Co je R
- R Markdown

### Použitá literatura

### Reference