

Dr. Racket

- racket-lang.org
- multiplatformní
- podpora několika jazyků → na začátku nutno vybrat jazyk: **Language / Choose Language / Intermediate Student** nebo vyšší v sekci Teaching Languages

Prostředí Dr. Racket

The screenshot shows the Dr. Racket IDE interface. The top window contains Racket code: `(define (nadruhou x) (* x x))` and `(nadruhou 5)`. The bottom window shows a welcome message and a prompt `>`. Red arrows point from text labels to specific parts of the interface: 'okno definic (program)' points to the code editor, 'ladění spustit' points to the 'Debug' and 'Run' buttons, and 'interpret (výrazy k vyhodnocení)' points to the prompt in the bottom window.

```
File Edit View Language Racket Insert Tabs Help
Untitled (define ...)
Check Syntax Debug Run Stop

(define (nadruhou x)
  (* x x)
)

(nadruhou 5)

Welcome to DrRacket, version 6.11 [3m].
Language: Advanced Student; memory limit: 128 MB.
>
```

okno definic (program)

ladění
spustit

interpret (výrazy k vyhodnocení)

Advanced Student 5:12 307.00 MB

Úloha 1

- definujte ve Scheme funkci (**prepona a b**) se dvěma argumenty, která vrací délku přepony pravoúhlého trojúhelníka s odvěsnami **a** a **b**
- např. (prepona 3 4) vrátí 5

Řešení 1

```
(define (prepona a b)
  (sqrt (+ (* a a)
           (* b b))))
```

Úloha 2

- vytvořte funkci (**pravouhly?** a b c), která vrátí, zda trojúhelník se stranami a , b a c je pravoúhlý (platí v něm Pythagorova věta)

Řešení 2

```
(define (pravouhly? a b c)
  (= (* c c)
     (+ (* a a)
        (* b b))))
```

Úloha 3

- vytvořte funkci **C->F**, která převádí teplotu ze stupňů Celsia na stupně Fahrenheita
- $\text{Fahrenheit} = \text{Celsius} * (9/5) + 32$
- např. **(C->F 20)** by mělo vrátit 68

Řešení 3

```
(define (C->F teplota)
  (+ (* teplota
       (/ 9 5))
     32))
```


Úloha 4

- vytvořte funkci **ab3**, která dostane parametry a a b a vypočítá $(a + b)^3$
- $(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$
- např. (**ab3 2 3**) by mělo vrátit 125

Řešení 4

```
(define (ab3 a b)
  (expt (+ a b) 3))
```

```
(define (ab3 a b)
  (* (+ a b) (+ a b) (+ a b)))
```

```
(define (ab3 a b)
  (+ (* a a a)
     (* 3 (* a a) b)
     (* 3 a (* b b))
     (* b b b)))
```

Úloha 5

- vytvořte funkci **skorocele?**, která vrátí, zda se její argument (reálné číslo) liší od nejbližšího celého čísla o méně než 0.001
- např. (skorocele? 7.99) by mělo vrátet #false, (skorocele? 7.9999) #true

Řešení 5

```
(define (skorocele? x)
  (< (abs (- x
            (round x)))
     0.001))
```