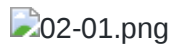


- [Richiami di nozioni base dei linguaggi formali](#)
    - [Sintassi delle espressioni regolari](#)
    - [Semantica delle espressioni regolari](#)
  - [Classificazione dei token dei linguaggi di programmazione](#)
- 

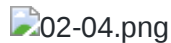
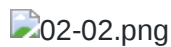
## Richiami di nozioni base dei linguaggi formali

---



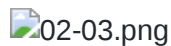
### Sintassi delle espressioni regolari

---



### Semantica delle espressioni regolari

---



Manca l'insieme vuoto.



---

## Classificazione dei token dei linguaggi di programmazione

---

- Parole chiave (keyword)
  - `if, then, else, while, ...`
  - Si usa l'alternanza: `if | then | else | while ...`
  - Unica categoria sintattica: **KEYWORD**
  - Case insensitive (e.g., SQL)
    - `select, SELECT, SeLeCt, SElect, ...`
    - Noioso e troppo verboso: si usano impostazioni ad hoc
- Identificatori
  - `[a-zA-Z_][0-9a-zA-Z_]*`
  - `[a-zA-Z_]([0-9] | [a-zA-Z_])*`
  - `DIGIT = [0-9]`
  - `LETTER = [a-zA-Z_]`
  - `{LETTER}({DIGIT}|{LETTER})*`
- Costanti letterali (interi, floating point, stringa, ecc.)
  - Costanti intere

- `{DIGIT}+`

- Nota: accetta `000000` , non accetta `-1`

- Costanti floating point

- `[+-]?[0-9]+.[0-9]*`

- occorre distinguere iterazione p