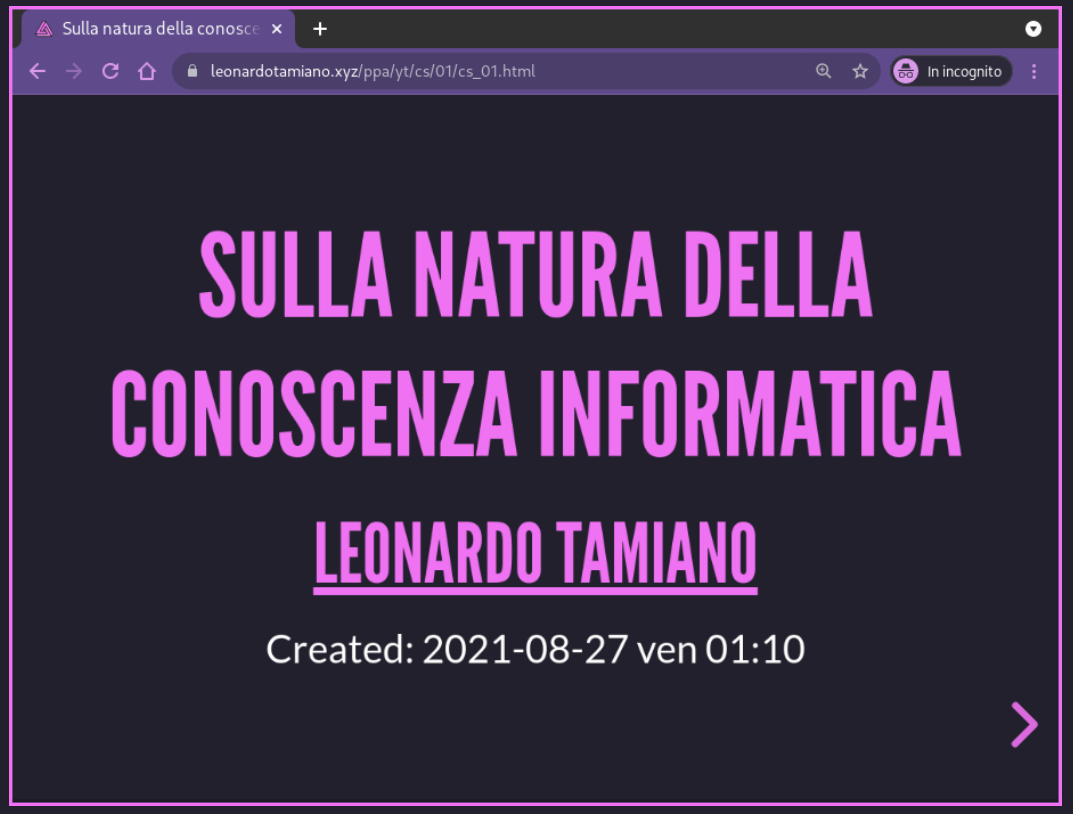
MACCHINE, LINGUAGGI E LA TESI DI CHURCH-TURING

TABLE OF CONTENTS

RIASSUNTO

Nella lezione precedente avevamo introdotto la **conoscenza procedurale**, che è la principale conoscenza studiata e portata avanti dall'informatica.



Per cercare di caratterizzare meglio questa conoscenza abbiamo trattato il problema del **calcolo della radice quadrata**

$$5 \longrightarrow \sqrt{5} \approx 2.2360679775$$

In generale,

$$x \overset{?}{\longrightarrow} \sqrt{x}$$

A tale fine è stato introdotto il **metodo di Netwon**, che può essere descritto come segue

```
Per calcolare la radice quadrata di x:
    Sia g = 1
    Sia precision = 0.001

    Ripeti fino a quando |g^2 - x| < precision:
       g = (g + x/g)/2

    Esci e ritorna g
```

Successivamente abbiamo fatto vedere come implementare tale metodo in tre diversi linguaggi di programmazione:

Python

```python
print("Hello World!")
```

C

```c
#include <stdio.h>

int main(int argc, char **argv) {
  printf("Hello World!");
  return 0;
}
```

Emacs-Lisp

```elisp
(message "Hello World!")
```

Iniziamo la nuova lezione dalla seguente domanda:

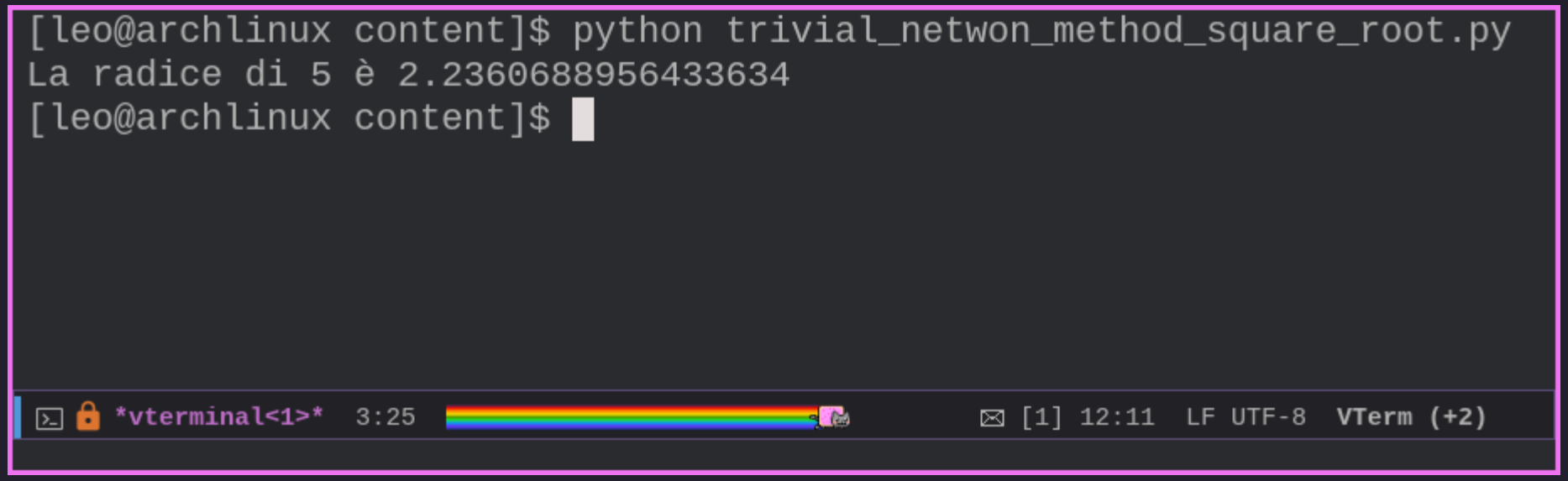
Che relazioni sussistono tra il linguaggio utilizzato per descrivere un algoritmo, e la macchina in grado di eseguire tale algoritmo?

LINGUAGGI E MACCHINE

La prima descrizione fornita del metodo di Netwon è stata scritta in **pseudocodice**, in quanto è:

1. Abbastanza precisa da permetterci di capire cosa fare passo passo.
2. Non abbastanza precisa da poter essere implementata ed eseguita direttamente su un computer digitale.

Per eseguire il metodo di Netwon sul nostro computer infatti abbiamo prima dovuto **tradurre** la descrizione data, in un vero e proprio linguaggio di programmazione.



Supponiamo però di avere solamente a disposizione lo pseudocodice.

Come possiamo eseguire il metodo di Netwon in questo caso particolare?

Come possiamo eseguire il metodo di Netwon?

L'idea è quella di prendere un foglio di carta, una penna, ed eseguire le istruzioni dello pseudocodice una alla volta, fino a quando non arriviamo alla condizione di terminazione.

Come possiamo eseguire il metodo di Netwon?

Per eseguire queste operazioni abbiamo bisogno di un qualcosa che coordina le varie attività necessarie al calcolo.

Nel caso degli esseri umani questo qualcosa è il nostro **cervello**.

Detto altrimenti,

È il cervello ad interpretare ed eseguire le istruzioni scritte in pseudocodice.

Per noi informatici quindi il cervello si comporta come un **modello di calcolo**, in quanto è una macchina che ci permette di eseguire degli algoritmi.

Osservazione: Il cervello è una macchina fisica, nel senso che esiste nel mondo reale. Non tutte le macchine però devono necessariamente essere fisiche. Ciò che conta è la descrizione matematica della macchina, ovvero il **modello di calcolo**.

Gli algoritmi eseguiti dal cervello possono essere scritti in **pseudocodice**.

Notiamo che **non esiste una sintassi specifica per lo pseudocodice**: il cervello infatti è una macchina molto sofisticata, in grado di capire se sequenze di parole sintatticamente diverse hanno o meno lo stesso significato.

In questo caso abbiamo che il linguaggio utilizzato per descrivere gli algoritmi è lo **pseudocodice**, e la macchina utilizzata per eseguire questi algoritmi è il nostro **cervello**.

$$\text{Pseudocodice} \longleftrightarrow \text{Cervello}$$

Ciò che abbiamo notato in questo caso particolare in realtà ha una natura molto più generale, e rappresenta un punto chiave da capire se si vuole comprendere bene l'informatica.

È di fondamentale importanza, nell'informatica, osservare le relazioni presenti tra questi due concetti

$$\text{Linguaggio} \longleftrightarrow \text{Macchina}$$

> *— Non parlo — dichiarò Bijaz. — Uso una macchina chiamata lingua. Cingola e grugnisce, ma mi appartiene.*
>
> *Messia di Dune, Frank Herbert.*

Linguaggio $\longleftarrow$ Macchina

Ogni macchina definisce un particolare linguaggio:

il linguaggio composto da tutte e sole le istruzioni che quella macchina può eseguire.

$\text{Linguaggio} \longrightarrow \text{Macchina}$

Viceversa, ogni linguaggio definisce una particolare macchina:

la macchina in grado di eseguire tutte e sole le istruzioni di quel particolare linguaggio.

Come abbiamo già detto, non tutte le macchine devono essere costruite fisicamente.

Consideriamo ad esempio il linguaggio **python**.

La macchina associata a tale linguaggio sarebbe troppo complessa, in termine di **circuiti elettrici**, da costruire fisicamente.

Consideriamo ad esempio il linguaggio **python**.

Per potere eseguire del codice **python** l'idea è quella di implementare tale macchina sotto forma di **software** tramite delle tecniche di **traduzione**.

Tramite la traduzione siamo in grado di prendere un programma python, ed eseguire ogni istruzione presente in tale programma.

Consideriamo ad esempio il linguaggio **python**.

In sostanza, l'idea è quella di **simulare il comportamento della macchina associata al linguaggio python con un'altra macchina**.

Consideriamo ad esempio il linguaggio **python**.

A differenza della macchina python però, **la macchina utilizzata per questa simulazione deve essere più semplice da costruire tramite dei circuiti elettrici**.

L'obiettivo dei prossimi video sarà quello di trattare in dettaglio le varie tecniche di **traduzione** (interpretazione, compilazione, etc.) e di far vedere come queste sono utilizzate nella costruzione dei moderni computer digitali.

Prima però è necessario menzionare un'idea molto importante, nota con il nome di **Tesi di Church-Turing**.

LA TESI DI CHURCH-TURING

Ricapitolando da quanto già detto,

- **Modello di calcolo**: descrizione formale di una macchina in grado di eseguire degli **algoritmi**.

- **Algoritmo**: sequenza di istruzioni elementari finita e non ambigua che può essere eseguita su un particolare **modello di calcolo**.

La parola "formale" in questo contesto indica il fatto che non dobbiamo descrivere tutti i dettagli del funzionamento interno della macchina, ma solo il suo funzionamento logico, ovvero ciò che permette di fare.

Nel corso degli anni sono stati definiti e sviluppati tantissimi modelli di calcolo:

- Macchine di Turing.
- Funzioni ricorsive.
- Lambda calcolo.
- Le macchine a registri elementari.
- I comuni linguaggi di programmazione.

Anche se questi formalismi sono molto diversi gli uni dagli altri, tutti questi linguaggi permettono di descrivere lo stesso concetto:

il concetto di computazione, ovvero il concetto di calcolo automatico.

Detto altrimenti, ciò che è calcolabile secondo uno di questi linguaggi, lo è anche secondo tutti gli altri.

La **tesi di Church-Turing** dice proprio questo:

È calcolabile tutto (e solo) ciò che può essere calcolato tramite una macchina di Turing.

È proprio questa equivalenza tra i diversi modelli di calcolo che permette di applicare nella pratica la tecnica della **traduzione**:

La macchina di python è, computazionalmente parlando, equivalente ad un'altra macchina, una più facile da costruire nella realtà.

PROSSIMAMENTE

Per dare un'idea concreta di modello di calcolo, nel prossimo video andremmo ad analizzare uno dei modelli più semplici ed intuitivi:

La macchina di Turing