

Corso di impiego industriale dell'energia



INTRODUZIONE A PYTHON

Prof. Rodolfo Taccani

Ing. Pivetta Davide

AA 2020 – 21



ANACONDA®

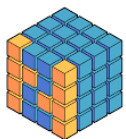
Introduzione di Python in ambiente Anaconda

Ambiente di lavoro adatto all'utilizzo del linguaggio di programmazione Python. Risulta uno degli ambienti più utilizzati per lo sviluppo di sistemi di calcolo.

VANTAGGI:

- Gestione rapida di librerie e strumenti;
- Sviluppo semplificato di sistemi di calcolo;
- Contiene strumenti per analisi dei dati e delle performance;
- Contiene pacchetti per rappresentare graficamente i risultati.

Introduzione di Python in ambiente Anaconda



Numpy: pacchetto per il calcolo scientifico (operazioni tra matrice, definizione funzioni, ecc.)



Pandas: è uno strumento veloce, potente e flessibile per l'analisi dei dati e la loro semplice manipolazione.



Matplotlib: è una libreria completa per creare visualizzazioni statiche, animate e interattive.



Spyder: strumento per la scrittura, analisi, correzione errori, ecc. Si tratta di un IDE («Integrated Development Environment») adatto alla risoluzione di problemi scientifici e ingegneristici.



Jupyter: è un ambiente di sviluppo interattivo web-based. Si adatta all'interfaccia utente per supportare un ampio range di flussi di lavoro.



Download Anaconda con Python

ANACONDA®

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Upgrade Now

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Applications on

base (root)

Channels

Refresh

<p>console_shortcut 0.1.1 Console shortcut creator for Windows (using menuinst)</p> <p>Launch</p>	<p>JupyterLab 1.1.4 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.</p> <p>Launch</p>	<p>Notebook 6.0.1 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p> <p>Launch</p>	<p>powershell_shortcut 0.0.1</p> <p>Launch</p>	<p>Qt Console 4.5.5 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.</p> <p>Launch</p>	<p>Spyder 3.3.6 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p> <p>Launch</p>
<p>Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets.</p> <p>Install</p>	<p>Orange 3 3.23.1 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.</p> <p>Install</p>	<p>RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.</p> <p>Install</p>	<p>VS Code 1.43.0 Streamlined code editor with support for development operations like debugging, task running and version control.</p> <p>Install</p>		

Documentation

Developer Blog





Download Anaconda con Python

ANACONDA®

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Upgrade Now

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Applications on base (root)

Channels

Refresh

The screenshot displays the 'Applications' section of the Anaconda Navigator interface. It features a grid of application cards, each with an icon, name, version, description, and a button to either 'Launch' or 'Install' the application. The applications shown are:

- console_shortcut** (0.1.1): Console shortcut creator for Windows (using menuinst). Button: Launch
- JupyterLab** (1.1.4): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Button: Launch
- Notebook** (6.0.1): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Button: Launch
- powershell_shortcut** (0.0.1): Button: Launch
- Qt Console** (4.5.5): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Button: Launch
- Spyder** (3.3.6): Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Button: Launch
- Glueviz** (0.15.2): Multidimensional data visualization across files. Explore relationships within and among related datasets. Button: Install
- Orange 3** (3.23.1): Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Button: Install
- RStudio** (1.1.456): A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Button: Install
- VS Code** (1.43.0): Streamlined code editor with support for development operations like debugging, task running and version control. Button: Install

QUI troverete i diversi ambienti di lavoro!



Download Anaconda con Python

QUI potete vedere, aggiornare e scaricare le librerie e strumenti di Python



Anaconda Navigator

File Help

Upgrade Now

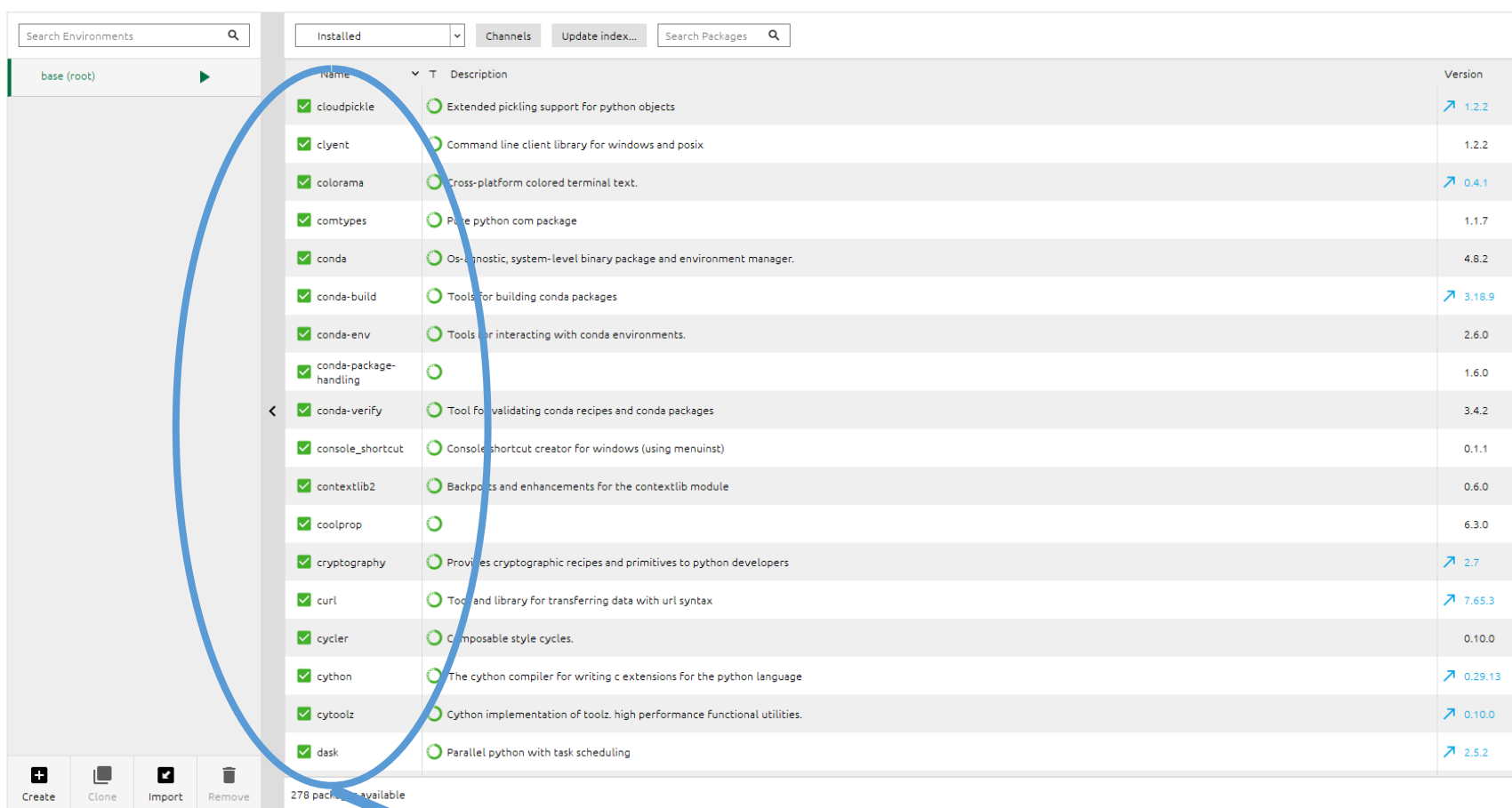
Sign in to Anaconda Cloud

- Home
- Environments**
- Learning
- Community
- Documentation
- Developer Blog
- Twitter
- YouTube
- LinkedIn

Application: base (root) Channels Refresh

 console_shortcut 0.1.1 Console shortcut creator for Windows (using menuinst) Launch	 JupyterLab 1.1.4 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 Notebook 6.0.1 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 powershell_shortcut 0.0.1 Launch	 Qt Console 4.5.5 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Launch	 Spyder 3.3.6 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features Launch
 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. Install	 Orange 3 3.23.1 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Install	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Install	 VS Code 1.43.0 Streamlined code editor with support for development operations like debugging, task running and version control. Install		

Download Anaconda con Python

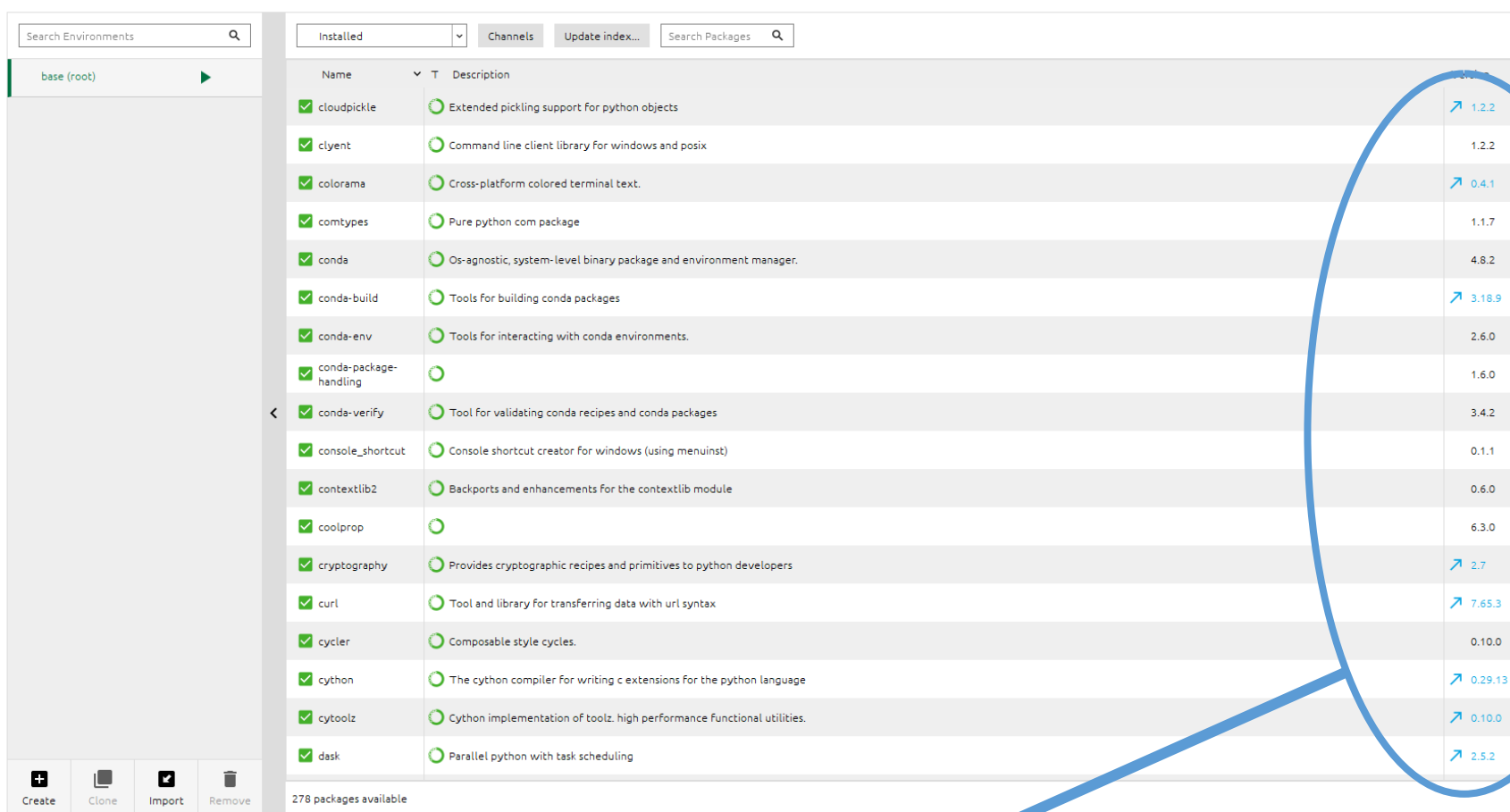


The screenshot shows the Anaconda environment manager interface. The left sidebar contains navigation options: Home, Environments, Learning, and Community. The main area displays a list of installed packages for the 'base (root)' environment. A blue oval highlights the 'Name' column, and a blue arrow points from it to the text below. The table lists various packages with their descriptions and versions.

Name	Description	Version
cloudpickle	Extended pickling support for python objects	1.2.2
clyent	Command line client library for windows and posix	1.2.2
colorama	Cross-platform colored terminal text.	0.4.1
comtypes	Pure python com package	1.1.7
conda	Os-agnostic, system-level binary package and environment manager.	4.8.2
conda-build	Tools for building conda packages	3.18.9
conda-env	Tools for interacting with conda environments.	2.6.0
conda-package-handling		1.6.0
conda-verify	Tool for validating conda recipes and conda packages	3.4.2
console_shortcut	Console shortcut creator for windows (using menuinst)	0.1.1
contextlib2	Backports and enhancements for the contextlib module	0.6.0
coolprop		6.3.0
cryptography	Provides cryptographic recipes and primitives to python developers	2.7
curl	Tool and library for transferring data with url syntax	7.65.3
cycler	Composable style cycles.	0.10.0
cython	The cython compiler for writing c extensions for the python language	0.29.13
cytoolz	Cython implementation of toolz: high performance functional utilities.	0.10.0
dask	Parallel python with task scheduling	2.5.2

QUI sono elencati i «pacchetti» già installati

Download Anaconda con Python



The screenshot displays the Anaconda environment manager interface. On the left, there is a sidebar with navigation options: Home, Environments, Learning, and Community. The main area shows a list of installed packages for the 'base (root)' environment. The packages are listed in a table with columns for Name, Description, and Version. A blue arrow points from the text below to the version column of the table.

Name	Description	Version
cloudpickle	Extended pickling support for python objects	1.2.2
clyent	Command line client library for windows and posix	1.2.2
colorama	Cross-platform colored terminal text.	0.4.1
comtypes	Pure python com package	1.1.7
conda	Os-agnostic, system-level binary package and environment manager.	4.8.2
conda-build	Tools for building conda packages	3.18.9
conda-env	Tools for interacting with conda environments.	2.6.0
conda-package-handling		1.6.0
conda-verify	Tool for validating conda recipes and conda packages	3.4.2
console_shortcut	Console shortcut creator for windows (using menuinst)	0.1.1
contextlib2	Backports and enhancements for the contextlib module	0.6.0
coolprop		6.3.0
cryptography	Provides cryptographic recipes and primitives to python developers	2.7
curl	Tool and library for transferring data with url syntax	7.65.3
cycler	Composable style cycles.	0.10.0
cython	The cython compiler for writing c extensions for the python language	0.29.13
cytoolz	Cython implementation of toolz: high performance functional utilities.	0.10.0
dask	Parallel python with task scheduling	2.5.2

QUI sono elencati i «pacchetti» già installati... e la loro versione

Consigli e link utili

COOLPROP:

- <http://www.coolprop.org/>
(Tutte le info necessarie)
- <http://www.coolprop.org/coolprop/HighLevelAPI.html#propssi-function>
(Funzione Propsi e tabella degli input)
- <http://coolprop.org/fluidproperties/PurePseudoPure.html#list-of-fluids>
(Fluidi presenti nella libreria)

Anaconda/Python:

- <https://conda-forge.org/>
- <https://github.com/>

Per eventuali domande: Davide Pivetta – davide.pivetta@phd.units.it



ANACONDA®

Utilizzo dell'IDE Spyder

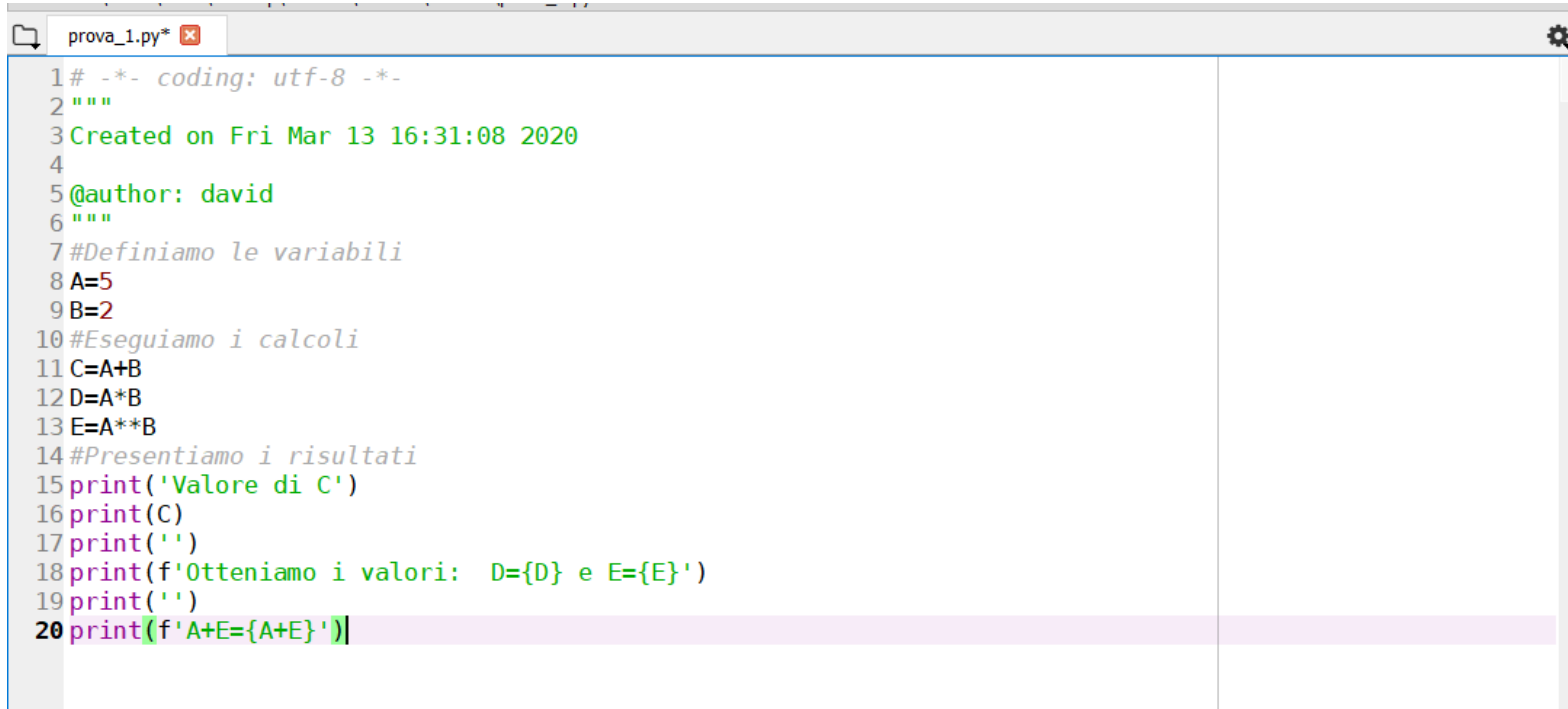
Ora apriamo Spyder

The screenshot shows the Anaconda Navigator interface. At the top, there's a navigation bar with 'File' and 'Help' menus, and buttons for 'Upgrade Now' and 'Sign in to Anaconda Cloud'. The main area displays a grid of application cards under the heading 'Applications on base (root) Channels'. The cards include:

- console_shortcut 0.1.1: Console shortcut creator for Windows (using menuinst). [Launch]
- JupyterLab 1.1.4: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. [Launch]
- Notebook 6.0.1: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. [Launch]
- powershell_shortcut 0.0.1: [Launch]
- Qt Console 4.5.5: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. [Launch]
- Spyder 3.3.6: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and visualization features. [Launch]
- Glueviz 0.15.2: Multidimensional data visualization across files. Explore relationships within and among related datasets. [Install]
- Orange 3 3.23.1: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. [Install]
- RStudio 1.1.456: A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. [Install]
- VS Code 1.43.0: Streamlined code editor with support for development operations like debugging, task running and version control. [Install]

At the bottom left, there are links for 'Documentation' and 'Developer Blog', and social media icons for Twitter, YouTube, and GitHub.

Andiamo a scrivere lo script:



```
1# -*- coding: utf-8 -*-
2"""
3Created on Fri Mar 13 16:31:08 2020
4
5@author: david
6"""
7#Definiamo le variabili
8A=5
9B=2
10#Eseguiamo i calcoli
11C=A+B
12D=A*B
13E=A**B
14#Presentiamo i risultati
15print('Valore di C')
16print(C)
17print('')
18print(f'Otteniamo i valori: D={D} e E={E}')
19print('')
20print(f'A+E={A+E}')
```

Andiamo a scrivere lo script:

```
prova_1.py*  
1# -*- coding: utf-8 -*-  
2"""  
3Created on Fri Mar 13 16:31:08 2020  
4  
5@author: david  
6"""  
7#Definiamo le variabili  
8A=5  
9B=2  
10#Eseguiamo i calcoli  
11C=A+B  
12D=A*B  
13E=A**B  
14#Presentiamo i risultati  
15print('Valore di C')  
16print(C)  
17print('')  
18print(f'Otteniamo i valori: D={D} e E={E}')19print('')  
20print(f'A+E={A+E}')
```

Nota:

- # : si usa per inserire dei commenti;

Andiamo a scrivere lo script:

```
prova_1.py*  
1# -*- coding: utf-8 -*-  
2"""  
3Created on Fri Mar 13 16:31:08 2020  
4  
5@author: david  
6"""  
7#Definiamo le variabili  
8A=5  
9B=2  
10#Eseguiamo i calcoli  
11C=A+B  
12D=A*B  
13E=A**B  
14#Presentiamo i risultati  
15print('valore di C')  
16print(C)  
17print('')  
18print(f'Otteniamo i valori: D={D} e E={E}')19print('')  
20print(f'A+E={A+E}')
```

Nota:

- # : si usa per inserire dei commenti;
- ** : operazione per elevare a potenza;

Andiamo a scrivere lo script:

```
prova_1.py*  
1# -*- coding: utf-8 -*-  
2"""  
3Created on Fri Mar 13 16:31:08 2020  
4  
5@author: david  
6"""  
7#Definiamo le variabili  
8A=5  
9B=2  
10#Eseguiamo i calcoli  
11C=A+B  
12D=A*B  
13E=A**B  
14#Presentiamo i risultati  
15print('Valore di C')  
16print(C)  
17print('')  
18print(f'Otteniamo i valori: D={D} e E={E}')  
19print('')  
20print(f'A+E={A+E}')
```

Nota:

- print() : «stampare» il risultato;
- print('Posso inserire all'interno qualsiasi frase o simbolo')

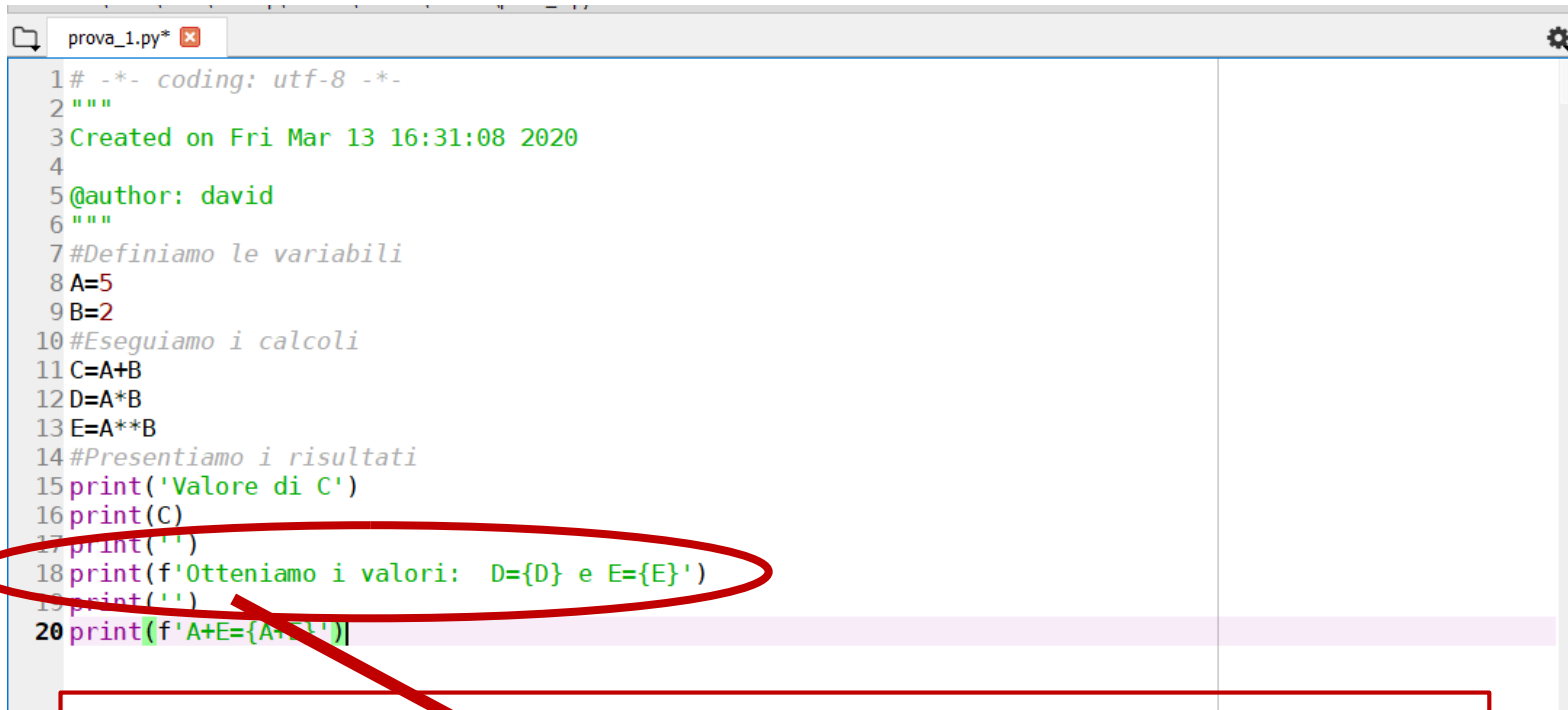
Andiamo a scrivere lo script:

```
prova_1.py*  
1# -*- coding: utf-8 -*-  
2"""  
3Created on Fri Mar 13 16:31:08 2020  
4  
5@author: david  
6"""  
7#Definiamo le variabili  
8A=5  
9B=2  
10#Eseguiamo i calcoli  
11C=A+B  
12D=A*B  
13E=A**B  
14#Presentiamo i risultati  
15print('Valore di C')  
16print(C)  
17print('')  
18print(f'Otteniamo i valori: D={D} e E={E}')  
19print('')  
20print(f'A+E={A+E}')
```

Nota:

- print() : «stampare» il risultato;
- print('Posso inserire all'interno qualsiasi frase o simbolo')
- print(" ") : inserisce uno spazio;

Andiamo a scrivere lo script:



```
1# -*- coding: utf-8 -*-
2"""
3Created on Fri Mar 13 16:31:08 2020
4
5@author: david
6"""
7#Definiamo le variabili
8A=5
9B=2
10#Eseguiamo i calcoli
11C=A+B
12D=A*B
13E=A**B
14#Presentiamo i risultati
15print('Valore di C')
16print(C)
17print('')
18print(f'Otteniamo i valori: D={D} e E={E}')
19print('')
20print(f'A+E={A+E}')
```

Nota:

- `print()` : «stampare» il risultato;
- `print('Posso inserire all'interno qualsiasi frase o simbolo')`
- `print('')` : inserisce una linea vuota;
- `print(f'')`: così all'interno posso scrivere una frase e la variabile da mostrare va inserita dentro `{}` ;



Utilizzo dell'IDE Spyder

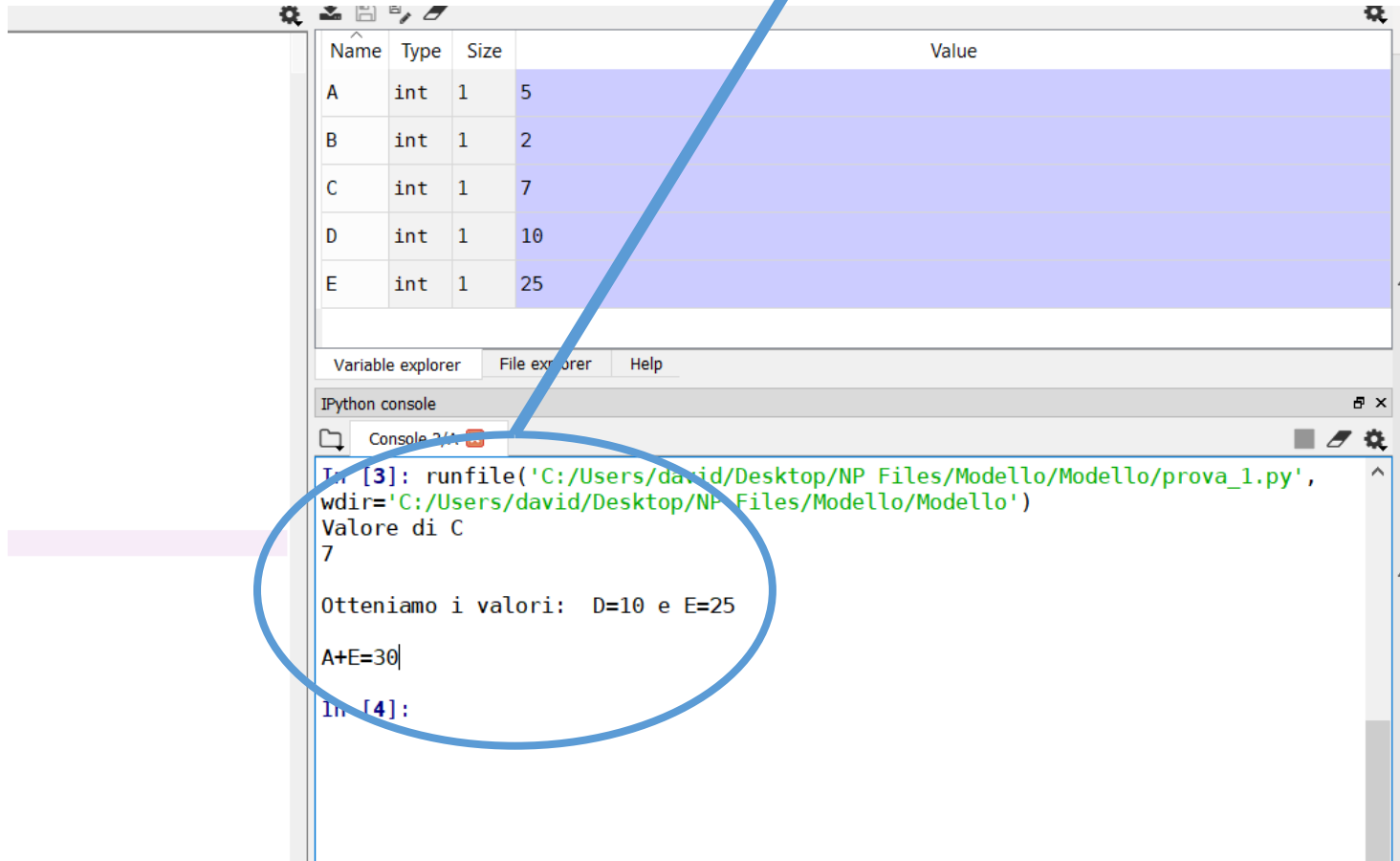
Avviamo la risoluzione dello script

The screenshot shows the Spyder IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The Run button, represented by a green play icon, is circled in blue. A blue arrow points from this button to the text 'Avviamo la risoluzione dello script'. The editor window displays a Python script named 'prova_1.py' with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Mar 13 16:31:08 2020
4
5 @author: david
6 """
7 #Definiamo le variabili
8 A=5
9 B=2
10 #Eseguiamo i calcoli
11 C=A+B
12 D=A*B
13 E=A**B
14 #Presentiamo i risultati
15 print('Valore di C')
16 print(C)
17 print('')
18 print(f'Otteniamo i valori: D={D} e E={E}')
19 print('')
20 print(f'A+E={A+E}')
```

Utilizzo dell'IDE Spyder

Otteniamo i risultati richiesti



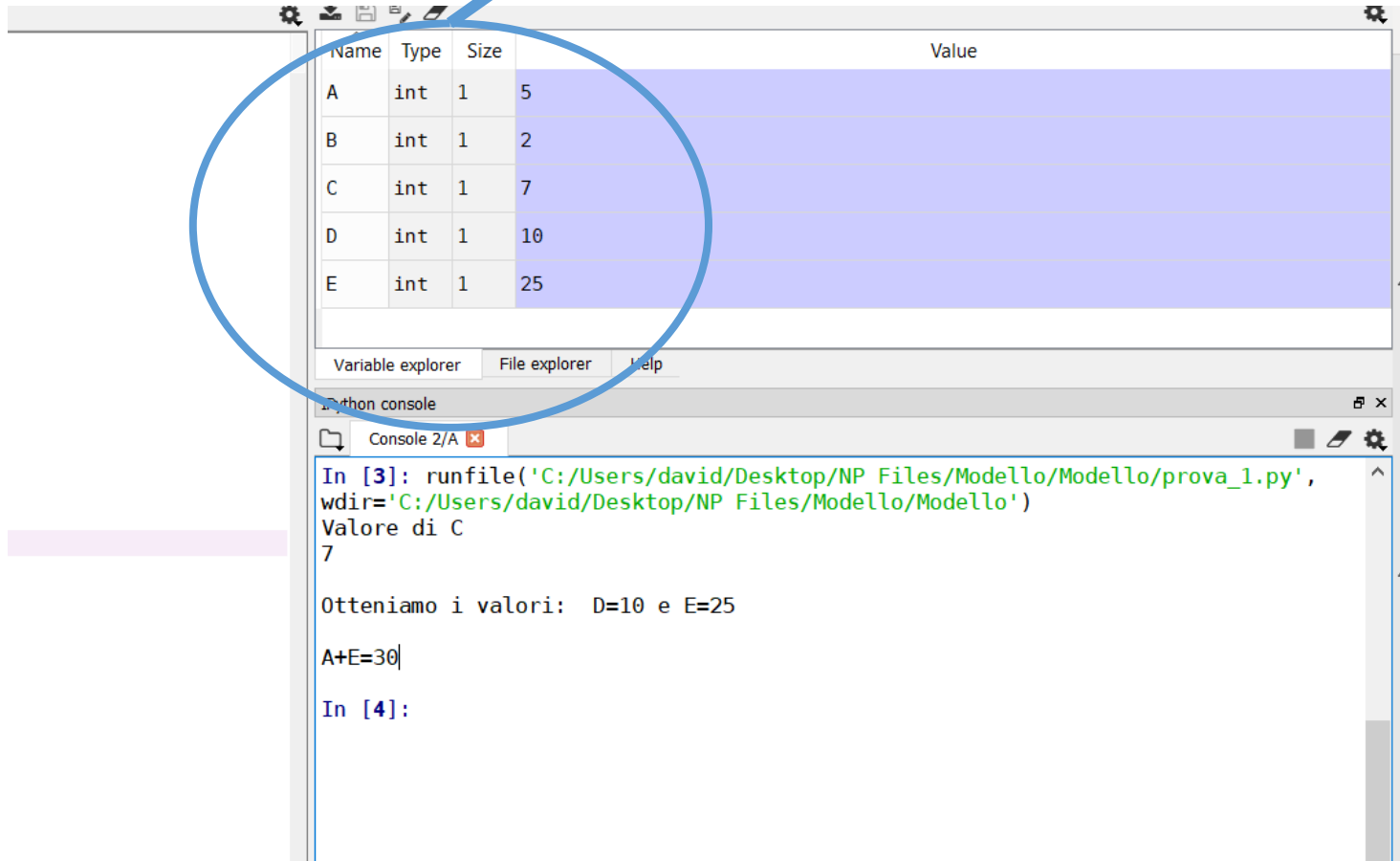
The screenshot shows the Spyder IDE interface. The top panel is the Variable explorer, which displays a table of variables. The bottom panel is the IPython console, which shows the execution of a Python script. A blue arrow points from the console output to the variable explorer table. A blue circle highlights the console output.

Name	Type	Size	Value
A	int	1	5
B	int	1	2
C	int	1	7
D	int	1	10
E	int	1	25

```
In [3]: runfile('C:/Users/david/Desktop/NP Files/Modello/Modello/prova_1.py',  
wdir='C:/Users/david/Desktop/NP Files/Modello/Modello')  
Valore di C  
7  
  
Otteniamo i valori: D=10 e E=25  
A+E=30|  
In [4]:
```

Utilizzo dell'IDE Spyder

Clicchiamo su «Variable explorer» e possiamo vedere il valore delle variabili del sistema



Name	Type	Size	Value
A	int	1	5
B	int	1	2
C	int	1	7
D	int	1	10
E	int	1	25

```
In [3]: runfile('C:/Users/david/Desktop/NP Files/Modello/Modello/prova_1.py',  
wdir='C:/Users/david/Desktop/NP Files/Modello/Modello')  
Valore di C  
7  
  
Otteniamo i valori: D=10 e E=25  
  
A+E=30|  
  
In [4]:
```

Esercizio 2: Introduzione a CoolProp

- 1) Aprite un nuovo script;
- 2) Salvatelo come «prova_2»;
- 3) Importare i pacchetti che dobbiamo utilizzare;
- 4) Scriviamo i dati del problema.

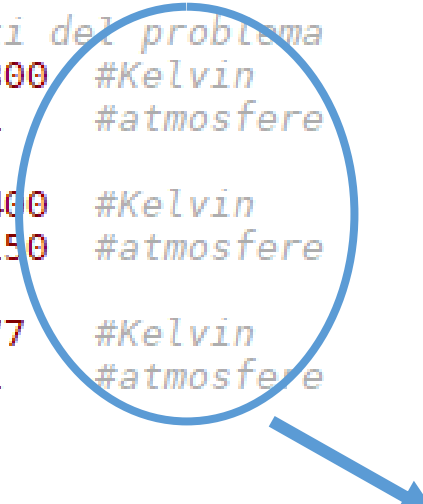
```
7
8 #Packages
9 from CoolProp import *
10
11 #Dati del problema
12 T1=300 #Kelvin
13 p1=1 #atmosfera
14
15 T2=400 #Kelvin
16 p2=150 #atmosfera
17
18 T3=77 #Kelvin
19 p3=1 #atmosfera
20
```

Importa da CoolProp tutte le funzioni disponibili

Esercizio 2: Introduzione a CoolProp

- 1) Aprite un nuovo script;
- 2) Salvatelo come «prova_2»;
- 3) Importare i pacchetti che dobbiamo utilizzare;
- 4) Scriviamo i dati del problema.

```
8 #Packages
9 from CoolProp.CoolProp import *
10
11 #Dati del problema
12 T1=300 #Kelvin
13 p1=1 #atmosfera
14
15 T2=400 #Kelvin
16 p2=150 #atmosfera
17
18 T3=77 #Kelvin
19 p3=1 #atmosfera
20
```



I commenti nel testo posso inserirli in qualsiasi punto della riga. Dopo il # il programma non legge più

Proprietà termodinamiche e unità di misura

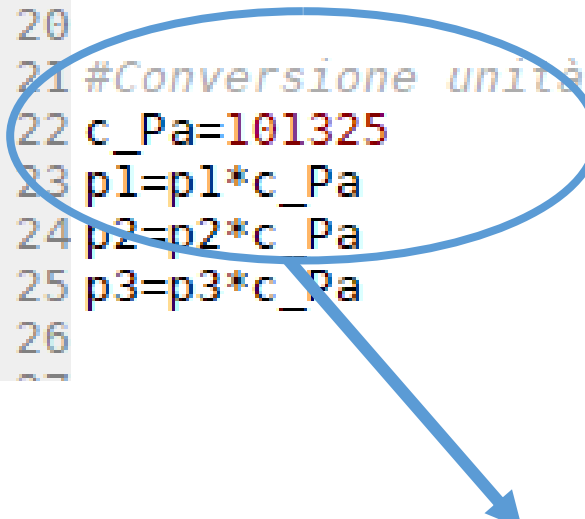
Parameter	Units	Input/Output	Trivial	Description
DELTA, Delta		IO	False	Reduced density (ρ/ρ_c)
DMOLAR, Dmolar	mol/m ³	IO	False	Molar density
D, DMASS, Dmass	kg/m ³	IO	False	Mass density
HMOLAR, Hmolar	J/mol	IO	False	Molar specific enthalpy
H, HMASS, Hmass	J/kg	IO	False	Mass specific enthalpy
P	Pa	IO	False	Pressure
Q	mol/mol	IO	False	Mass vapor quality
SMOLAR, Smolar	J/mol/K	IO	False	Molar specific entropy
S, SMASS, Smass	J/kg/K	IO	False	Mass specific entropy
TAU, Tau		IO	False	Reciprocal reduced temperature (T_c/T)
T	K	IO	False	Temperature
UMOLAR, Umolar	J/mol	IO	False	Molar specific internal energy
U, UMASS, Umass	J/kg	IO	False	Mass specific internal energy
ACENTRIC, acentric		O	True	Acentric factor
ALPHA0, alpha0		O	False	Ideal Helmholtz energy
ALPHAR, alphas		O	False	Residual Helmholtz energy
A, SPEED_OF_SOUND, speed_of_sound	m/s	O	False	Speed of sound
BVIRIAL, Bvirial		O	False	Second virial coefficient

<http://www.coolprop.org/coolprop/HighLevelAPI.html#propssi-function>

Esercizio 2: Introduzione a CoolProp

- 5) CoolProp richiede le pressioni in Pascal. Andiamo a convertire i nostri dati;

```
20  
21 #Conversione unità  
22 c_Pa=101325  
23 p1=p1*c_Pa  
24 p2=p2*c_Pa  
25 p3=p3*c_Pa  
26  
27
```



Python è compilativo, quindi «legge dall'alto verso il basso». In questo punto noi modifichiamo il valore della variabile p1, p2, p3.

Esercizio 2: Introduzione a CoolProp

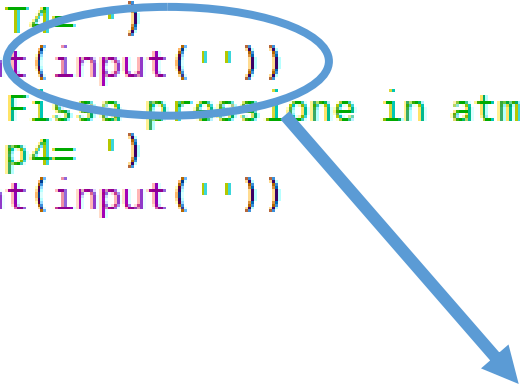
6) Vado a fissare le condizioni 4;

```
27 #Fisso punto 4
28 print('Fissa temperatura in Kelvin')
29 print('T4= ')
30 T4=float(input(''))
31 print('Fissa pressione in atmosfere')
32 print('p4= ')
33 p4=float(input(''))
34
```


Esercizio 2: Introduzione a CoolProp

6) Vado a fissare le condizioni 4;

```
27 #Fisso punto 4
28 print('Fissa temperatura in Kelvin')
29 print('T4= ')
30 T4=float(input(''))
31 print('Fissa pressione in atmosfere')
32 print('p4= ')
33 p4=float(input(''))
34
```



**Permette di inserire i valori richiesti
direttamente nella console**

Esercizio 2: Introduzione a CoolProp

6) Vado a fissare le condizioni 4;

```
27 #Fisso punto 4
28 print('Fissa temperatura in Kelvin')
29 print('T4= ')
30 T4=float(input(''))
31 print('Fissa pressione in atmosfere')
32 print('p4= ')
33 p4=float(input(''))
34
```

Converte la variabile «stringa» in variabile numerica decimale («float»)

Esercizio 2: Introduzione a CoolProp

- 7) Funzione «PropsSI» di CoolProp;
- 8) Calcolo le proprietà termodinamiche nelle condizioni 1;

```
19 #Calcolo proprietà termodinamiche
20 h1=PropsSI('H', 'T', T1, 'P', p1, 'Nitrogen')
21
22 s1=PropsSI('S', 'T', T1, 'P', p1, 'Nitrogen')
```

Proprietà che voglio calcolare

Proprietà Termodinamica	Simbolo	Unità di misura
Entalpia	H	J/kg
Entropia	S	J/kg/K
Temperatura	T	K
Pressione	P	Pa

Esercizio 2: Introduzione a CoolProp

- 7) Funzione «PropsSI» di CoolProp;
- 8) Calcolo le proprietà termodinamiche nelle condizioni 1;

```
19 #Calcolo proprietà termodinamiche
20 h1=PropsSI('H', 'T', T1, 'P', p1, 'Nitrogen')
21
22 s1=PropsSI('S', 'T', T1, 'P', p1, 'Nitrogen')
```

Fisso la temperatura

Proprietà Termodinamica	Simbolo	Unità di misura
Entalpia	H	J/kg
Entropia	S	J/kg/K
Temperatura	T	K
Pressione	P	Pa

Esercizio 2: Introduzione a CoolProp

- 7) Funzione «PropsSI» di CoolProp;
- 8) Calcolo le proprietà termodinamiche nelle condizioni 1;

```
19 #Calcolo proprietà termodinamiche
20 h1=PropsSI('H', 'T', T1, 'P', p1, 'Nitrogen')
21
22 s1=PropsSI('S', 'T', T1, 'P', p1, 'Nitrogen')
```

Fisso la pressione

Proprietà Termodinamica	Simbolo	Unità di misura
Entalpia	H	J/kg
Entropia	S	J/kg/K
Temperatura	T	K
Pressione	P	Pa

Esercizio 2: Introduzione a CoolProp

- 7) Funzione «PropsSI» di CoolProp;
- 8) Calcolo le proprietà termodinamiche nelle condizioni 1;

```
19 #Calcolo proprietà termodinamiche
20 h1=PropsSI('H', 'T', T1, 'P', p1, 'Nitrogen')
21
22 s1=PropsSI('S', 'T', T1, 'P', p1, 'Nitrogen')
```

Scelgo il fluido che voglio trattare

Proprietà Termodinamica	Simbolo	Unità di misura
Entalpia	H	J/kg
Entropia	S	J/kg/K
Temperatura	T	K
Pressione	P	Pa

Fluidi di CoolProp

All the fluids included in CoolProp

Name	EOS	c_{p0}	λ	η	melt	σ
1-Butene	[17]					[18]
Acetone	[19]					[18]
Air	[10]		[20]	[20]		
Ammonia	[21]		[22]	[23]		[18]
Argon	[24]		[20]	[20]	[24]	[18]
Benzene	[25]		[26]	[27]		[18]
CarbonDioxide	[28]		[29]	[30]	[28]	[18]
CarbonMonoxide	[19]				[31]	[18]
CarbonylSulfide	[19]					[18]
CycloHexane	[32]			[33]	[34]	[18]
CycloPropane	[35]	[35]				[36]
Cyclopentane	[37]		[38]	[39]		[36]
D4	[40]					[36]
D5	[41]					[36]
D6	[42]					[36]
Deuterium	[43]					[18]
Dichloroethane	[40]					
DiethylEther	[44]					[36]

http://www.coolprop.org/fluid_properties/PurePseudoPure.html#list-of-fluids

Esercizio 2: Introduzione a CoolProp

9) Calcoliamo proprietà in tutti i punti;

```
35
36
37 #Calcolo entalpia
38 h1=PropsSI('H', 'T', T1, 'P', p1, 'N2') #J/kg
39 h2=PropsSI('H', 'T', T2, 'P', p2, 'N2')
40 h3=PropsSI('H', 'T', T3, 'P', p3, 'N2')
41 h4=PropsSI('H', 'T', T4, 'P', p4*c_Pa, 'N2')
42
43 #Calcolo entropia
44 s1=PropsSI('S', 'T', T1, 'P', p1, 'N2') #J/kg/K
45 s2=PropsSI('S', 'T', T2, 'P', p2, 'N2')
46 s3=PropsSI('S', 'T', T3, 'P', p3, 'N2')
47 s4=PropsSI('S', 'T', T4, 'P', p4*c_Pa, 'N2')
48
```


Esercizio 2: Introduzione a CoolProp

10) Presentiamo i risultati;

```
50
51 print('')
52 print(f"h1= {h1} [J/kg]")
53 print('')
54 print(f"s1= {s1/1000} [kJ/kg/K]")
55 print('')
56 print(f'h4= {int(h4)/1000} [kJ/kg]')
57 print('')
58 print(f's4= {"%.2f" % (s4/1000)} [kJ/kg/K]')
59
```

Esercizio 2: Introduzione a CoolProp

10) Presentiamo i risultati;

```
50
51 print('')
52 print(f"h1= {h1} [J/kg]")
53 print('')
54 print(f"s1= {s1/1000} [kJ/kg/K]")
55 print('')
56 print(f'h4= {int(h4)/1000} [kJ/kg]')
57 print('')
58 print(f's4= {"%.2f" % (s4/1000)} [kJ/kg/K]')
59
```



Converte il numero in un intero

Esercizio 2: Introduzione a CoolProp

10) Presentiamo i risultati;

```
50
51 print('')
52 print(f"h1= {h1} [J/kg]")
53 print('')
54 print(f"s1= {s1/1000} [kJ/kg/K]")
55 print('')
56 print(f'h4= {int(h4)/1000} [kJ/kg]')
57 print('')
58 print(f's4= {"%.2f" % (s4/1000)} [kJ/kg/K]')
59
```



"%.2f" % (a)

**Approssimazione a due cifre
decimali di «a»**

Esercizio 3: Introduzione numpy e matplotlib

- 1) File, «save copy as» e chiamarlo «prova 3»;
- 2) Aggiungiamo il pacchetto «numpy» e «matplotlib»;

```
1 #Packages
2 from CoolProp.CoolProp import *
3 import numpy as np
4 import matplotlib.pyplot as plt
5
```

- 3) Fisso la temperatura e creo vettore per la pressione con numpy;

```
13 #Dati del problema
14 T1=300 #Kelvin
15
16 T2=400 #Kelvin
17
18 T3=77 #Kelvin
19
20 #Pressione
21 p=np.arange(1,400,10)
```

Esercizio 3: Introduzione numpy e matplotlib

4) Creo dei vettori vuoti per inserire h e s calcolati;

```
#Creo vettori vuoti  
array_h=[]  
array_s=[]
```

5) Utilizzo un ciclo if per selezionare la temperatura voluta;

```
29  
30 print('A che temperatura vuoi operare?')  
31 A=input('')  
32 if A==1:  
33     T=T1  
34 elif A==2:  
35     T=T2  
36 elif A==3:  
37     T=T3  
38 else:  
39     print('Non hai scelto tra le temperature proposte')  
40
```

Esercizio 3: Introduzione numpy e matplotlib

6) Utilizzo un ciclo for per calcolare h e s al variare della pressione;

```
for temp in range(0, len(p)):  
    h=PropsSI('H', 'T', T, 'P', p[temp]*c_Pa, 'N2')    #J/kg  
    array_h.append(h/1000)  
    s=PropsSI('S', 'T', T, 'P', p[temp]*c_Pa, 'N2')    #J/kg/K  
    array_s.append(s/1000)
```

«Appende» il risultato ad un vettore

Esercizio 3: Introduzione numpy e matplotlib

7) Creiamo i grafici con matplotlib;

```
#Grafico dell'entalpia  
plt.plot(p,array_h,'r--')  
plt.axis([0, max(p),min(array_h),max(array_h)])  
plt.xlabel('Pressione (atm)')  
plt.ylabel('Entalpia (kJ/kg)')  
plt.show()
```

```
#Grafico dell'entropia  
plt.plot(p,array_s, 'b--')  
plt.axis([0, max(p),min(array_s),max(array_s)])  
plt.xlabel('Pressione (atm)')  
plt.ylabel('Entropia (kJ/kg/K)')  
plt.show()
```

Grazie per l'attenzione!



Consigli e link utili

COOLPROP:

- <http://www.coolprop.org/>
(Tutte le info necessarie)
- <http://www.coolprop.org/coolprop/HighLevelAPI.html#propssi-function>
(Funzione Propssi e tabella degli input)
- <http://coolprop.org/fluidproperties/PurePseudoPure.html#list-of-fluids>
(Fluidi presenti nella libreria)

Anaconda/Python:

- <https://conda-forge.org/>
- <https://github.com/>

Per eventuali domande: Davide Pivetta – davide.pivetta@phd.units.it