

.NET Interactive

piattaforma opensource per
insegnare computer science e
data science

Dr Diego Colombo



Imparare esplorando

E' fondamentale offrire agli studenti l'opportunita` di esplorare i concetti esposti a lezione mantenendo il centro dell'attenzione sulla lezione svolta

L'esplorazione autonoma facilita l'assorbimento delle astrazioni introdotte a lezione, e rende piu` personalizzato il processo di apprendimento

Essere in grado di integrare un'applicazione pratica immediatamente dopo l'introduzione di un concetto rende l'usufruizione di materiale didattico molto piu` localizzata e ne riduce la dispersivita`

Se gli Ebook hanno aperto la via al consumo piu` immediato e facile dei testi di riferimento sono state le esperienze Notebook ad integrare la parte pratica.

Cosa `e un Notebook

In generale I Notebook sono strumenti che consentono di rappresentare non solo materiale statico.

I notebook possono contenere celle con codice da eseguire. La valutazione di istruzioni puo` produrre material piu` ricco e complesso di semplice testo.

Markdown

Rendering Scatter plots

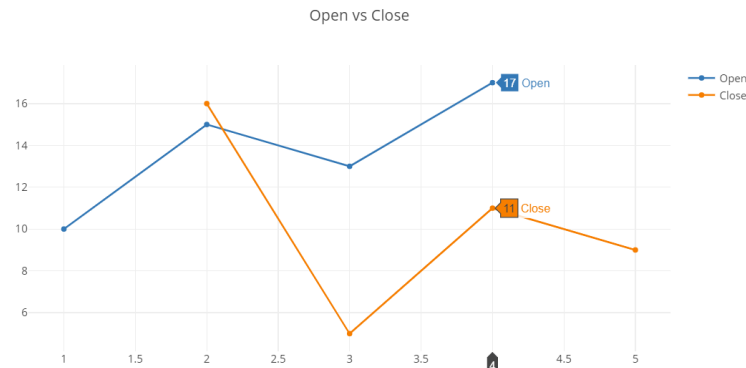
One of the most commonly used type of chart to explore data set. Use the type `Scatter`.

```
[3]: var openSeries = new Graph.Scatter
{
  name = "Open",
  x = new[] { 1, 2, 3, 4 },
  y = new[] { 10, 15, 13, 17 }
};

var closeSeries = new Graph.Scatter
{
  name = "Close",
  x = new[] { 2,3,4,5 },
  y = new[] { 16, 5, 11, 9 }
};

var chart = Chart.Plot(new[] { openSeries, closeSeries });
chart.WithTitle("Open vs Close");
display(chart);
```

Codice



Visualizzazione Interattiva

Demo

L'ecosistema dei notebook

I notebook sono molto diffusi e ce ne sono varie implementazioni.

Le piu` conosciute sono

- Matlab
- Mathematica
- Jupyter

La piattaforma Jupyter e` delle piu` adottate, soprattutto in campo scientifico, perche` e` open source e supportata da una vasta ed attiva comunita` che abbraccia sia l'accademia che l'industria

Esecuzione di un Notebook

Un Notebook si rappresenta e memorizza come file di testo.

Le celle che contengono codice sono associate ad un Kernel che e' in grado in interpretarne il contenuto e produrre output.

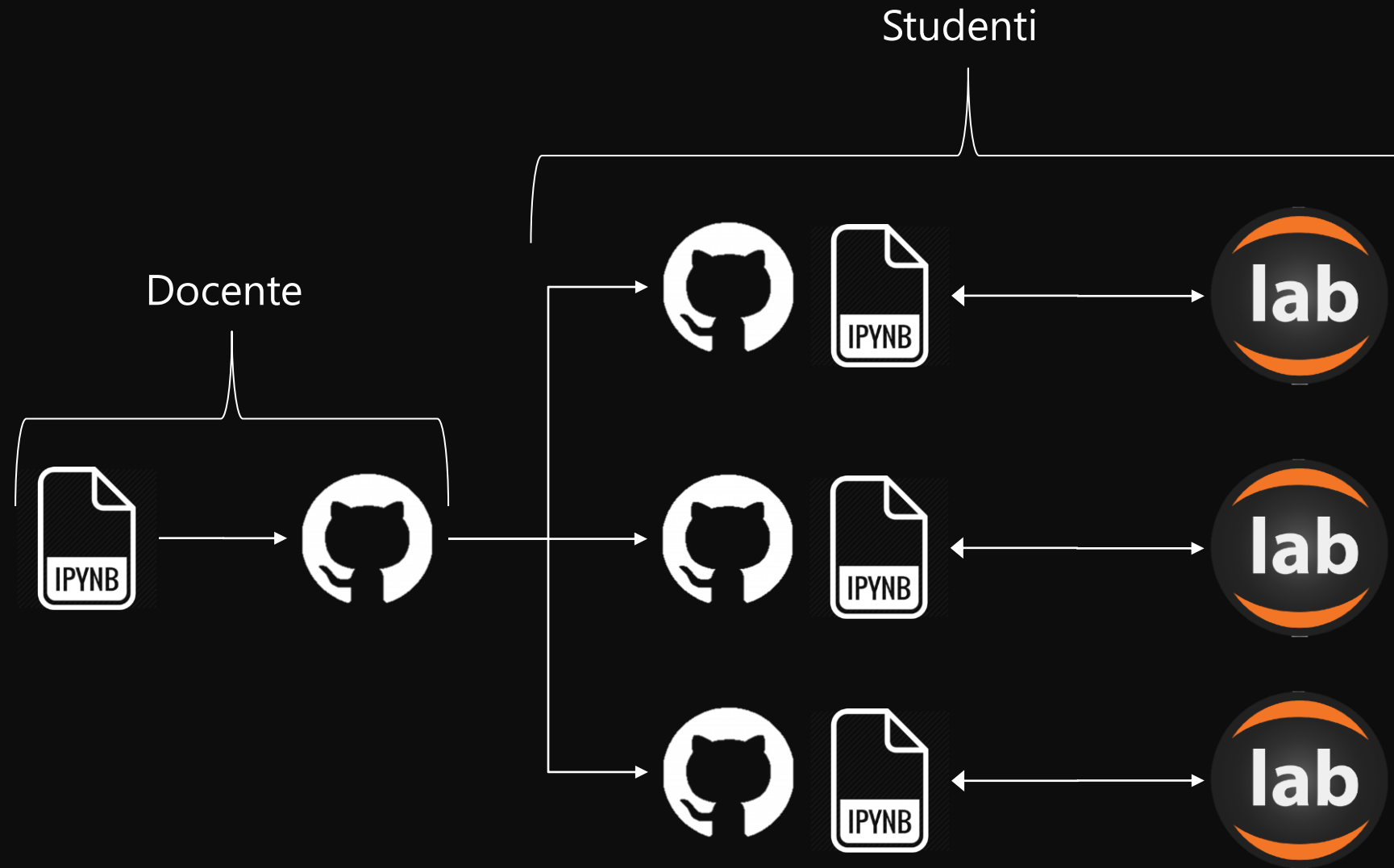
Le celle di output sono rappresentate come coppia contenuto e formato, facilitandone la portabilita`.

I kernel eseguono al di fuori del processo che visualizza I notebook e questa architettura rende aperta la scelta di client utilizzabili

.NET interactive

- Offre la possibilità di usare contemporaneamente diversi linguaggi di programmazione
- Permette al notebook di contenere definizione delle dipendenze
- Può eseguire in diverse topologie
- Può essere usato con diversi client
- Può essere integrato in sistemi
- Può essere esteso a tempo di esecuzione
- È completamente open source

Workflow Generale



Esecuzione cloud

Eseguire gli ambienti degli studenti in cloud / containers

- Garantisce la riproducibilità dell'elaborato
- Semplifica i costi di manutenzione e test
- Scarica ateneo e studente dalla necessità di avere risorse adeguate al tipo di progetto

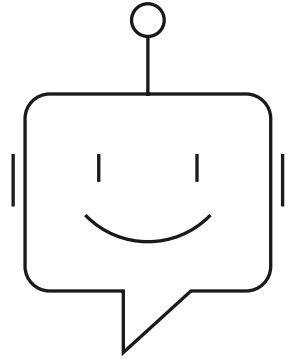
Licenze e controllo

Tutti i componenti necessari sono di pubblico dominio e sotto licenza opensource

- Runtime e SDK
 - .NET
 - Python
 - R
 - Julia
- Packaging
 - Docker containers
- UI / IDE
 - Jupyter
 - Jupyter lab
 - Nteract
 - Visual Studio Code
- Tooling
 - Repo2container (bynder project)
 - Binder

Risorse

- .NET Interactive (<https://www.github.com/dotnet/interactive>)
- Try .NET (<https://www.github.com/dotnet/try>)
- VsCode (<https://code.visualstudio.com/>)
- GitHub (<https://www.github.com>)
- GitHub Code Spaces
(<https://visualstudio.microsoft.com/services/github-codespaces/>)
- Creare un repository binder
(<https://github.com/dotnet/interactive/blob/main/docs/CreateBinder.md>)
- Notebook presentati(<https://github.com/colombod/interactive-notebooks>)



Thank you

Questions : diego.colombo@microsoft.com