Y qrhtco Ee qo r wvcvkqpcnP qvgdqqmf
Vj kpi uBqB/tÀ

This document is a live "notebook" that mixes text and code.

Run any piece of code by clicking inside the code, then pressing SHIFT + ENTER.

Let's start with something very easy:
(just click in the code below and press SHIFT + ENTER)
↓

```
2+2
```

Here's another computation:

```
1000!
```

Make a graphic:
(and try rotating it around)

```
Graphics3D[{Sphere[{0,0,0}],Sphere[{1,1,1}]}]
```

Here's another:
(recognize this one?)

```
PolyhedronData["RhombicHexecontahedron"]
```

Create a random network:

```
RandomGraph[{100,200}]
```

Do an analysis:
(% stands for the previous output)

```
CommunityGraphPlot[%]
```

Where does the cloud think you are?
(this is usually based on your IP address)

```
Here
```

Generate a map with a 5-mile-radius disk around there:

```
GeoGraphics[GeoDisk[Here,Quantity[5,"Miles"]]]
```

List the 5 nearest cities:

```
near=GeoNearest["City",Here,5]
```

Find their populations:

```
EntityValue[near,"Population"]
```

Make a pie chart:

```
PieChart[%,ChartLabels→near]
```

Generate a list of planets:

```
planets=PlanetData[]
```

Show an image for each planet:

```
images=EntityValue[planets,"Image"]
```

Detect edges in each image using image processing:

```
Map[EdgeDetect,images]
```

Create a web form where you can enter the name of any country:
(click the link to see the deployed version; it's a "Smart Field" that lets you use natural language)

```
CloudDeploy[FormFunction[{"country"→"Country"},Show[#country["Flag"],ImageS
```

Make a 3D plot of the terrain in a 5-mile radius around Mount Everest:

```
ListPlot3D [GeoElevationData [GeoBoundingBox [
    GeoDisk [Entity ["Mountain ", "MountEverest "], Quantity [5, "Miles "]]],
  GeoZoomLevel → 8], MeshFunctions → {#3 &},
 Filling → Bottom , ColorFunction → "Rainbow "]
```

Now create a function for doing this for any mountain, with any color scheme:

```
mountain[m_] :=Manipulate[ListPlot3D[
GeoElevationData[GeoBoundingBox[GeoDisk[m,    Quantity[5,"Miles"]]],GeoZoomLevel-
MeshFunctions  -> {#3 &}, Filling  -> Bottom,ColorFunction  -> ColorData[color
ImageSize->600,BoxRatios->{1,1,scaling}],    {colors,  ColorData["Gradients"]},{{s
```

Now deploy a form to run this from the web:

```
CloudDeploy [FormFunction [{"mountain " → "Mountain "},
  mountain [#mountain ] &, "CloudCDF "], Permissions → "Public "]
```

For more to try, see:

Wolfram Language Code Gallery »

Wolfram Language Documentation »

In[22]:=
```
f[x_] := x ^ 3 - 5 x + 1;
```

In[23]:=
```
a = 0 ;
```

In[24]:=
```
b = 1;
```

In[25]:=
```
ϵ = 0.01 ;
```

In[26]:=
```
NMax = 5;
```

In[29]:=
```
If[f[a] * f[b] > 0 , Print[
    "these values do not satisfy the IVP so change the initial value "],
  For[i = 1, i < NMax , i++, c = (a + b) / 2;
If[Abs[(b - a) / 2 ] < ϵ, Return [c],
Print[i, "th iteration value is : ", c];
Print["estimated error in ", i, "th iteration is: ", (b - a) / 2];
If[f[a] * f[c] < 0, b = c, a = c]]]];
Plot[f[x], {x, 0, 1}]
```

1th iteration value is : $\frac{7}{32}$

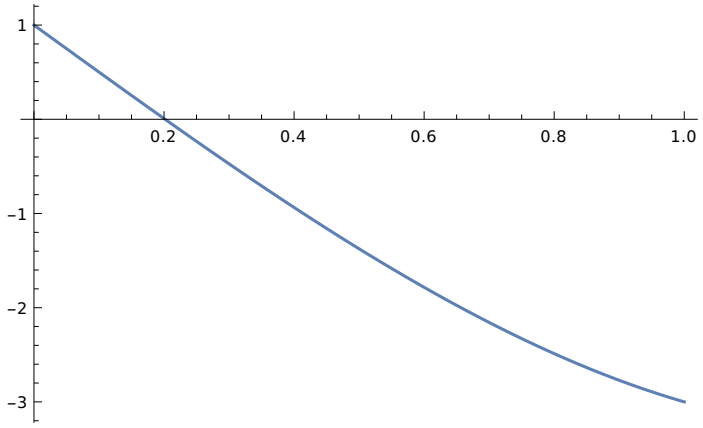
estimated error in 1th iteration is: $\frac{1}{32}$

2th iteration value is : $\frac{13}{64}$

estimated error in 2th iteration is: $\frac{1}{64}$

Out[29]= $\frac{25}{128}$

Out[30]=



In[31]:=
```
ClearAll
```

Out[31]= ClearAll

In[32]:=
```
f[x_] := x^3 + x^2 + x;
```

In[33]:=
```
a = -1;
```

In[34]:=
```
b = 1;
```

In[35]:=
```
ϵ = 0.01;
```

In[36]:=
```
NMax = 5;
```

In[37]:=
```
If[f[a] * f[b] > 0 , Print[
    "these values do not satisfy the IVP so change the initial value "],
  For[i = 1, i < NMax , i++, c = (a + b) / 2;
If[Abs[(b - a) / 2 ] < ϵ, Return[c],
Print[i, "th iteration value is : ", c];
Print["estimated error in ", i, "th iteration is: ", (b - a) / 2];
If[f[a] * f[c] < 0, b = c, a = c]]]];
Plot[f[x], {x, -1, 1}]
```

1th iteration value is : 0

estimated error in 1th iteration is: 1

2th iteration value is : $\frac{1}{2}$

estimated error in 2th iteration is: $\frac{1}{2}$

3th iteration value is : $\frac{3}{4}$

estimated error in 3th iteration is: $\frac{1}{4}$

4th iteration value is : $\frac{7}{8}$

estimated error in 4th iteration is: $\frac{1}{8}$

Out[38]=