Y qnhtco Eqo rwvcvkqpcn Pqvgdqqm

Vj kpi uBvqBVtÀ

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 »

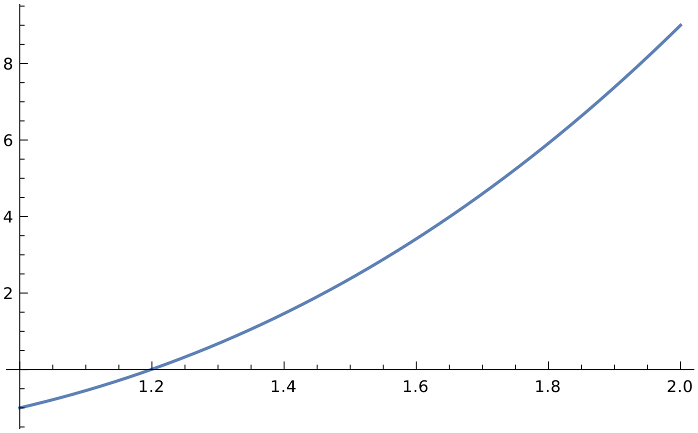
Wolfram Language Documentation »

```
In[10]:=  f [x_] := x ^ 3 + 2 x ^ 2 - 3 x - 1 ;

In[11]:=  Subscript [p, 0] = 2;

In[12]:=  Subscript [p, 1] = 1;

In[13]:=  ϵ = 0.000005 ;

In[14]:=  Nmax  = 10;

In[17]:=  For[n = 2, n ≤ Nmax , n++ ,
          Subscript [p, n] = N[Subscript [p, n - 1] -
                f[Subscript [p, n - 1]] (Subscript [p, n - 1] - Subscript [p, n - 2]) /
                  (f[Subscript [p, n - 1]] - f[Subscript [p, n - 2]])];
          If[Abs[Subscript [p, n] - Subscript [p, n - 1]] < ϵ, Return [Subscript [p, n]]];
          Print [n - 1, "th  iteration  value  is ", Subscript [p, n]];
          Print ["Estimated  error  is  :",
             Abs[Subscript [p, n] - Subscript [p, n - 1]]]];
          Plot [f[x], {x, 1, 2}]
          DiscretePlot [Subscript [p, i], {i, 0, 10}]
```

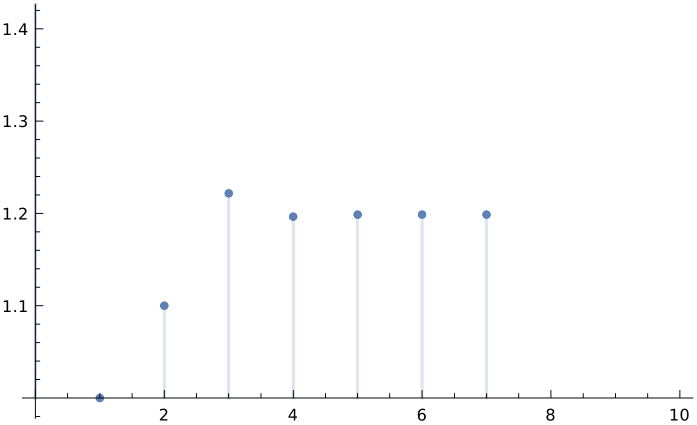
```
1th iteration  value  is 1.1

Estimated  error  is :0.1

2th iteration  value  is 1.22173

Estimated  error  is :0.121729

3th iteration  value  is 1.19649

Estimated  error  is :0.0252442

4th iteration  value  is 1.19865

Estimated  error  is :0.00216004

5th iteration  value  is 1.19869

Estimated  error  is :0.000045968
```

Out[17]= 1.19869

Out[18]=



Out[19]=


1.4
1.3
1.2
1.1
2
4
6
8
10