3.- CONTINUIDAD DE UNA FUNCIÓN REAL DE VARIABLE REAL PRELIMINARES TEÓRICOS

Continuidad de una función en un punto

Estudio de la continuidad de una función en un punto

Para estudiar la continuidad de una función en un punto debemos calcular la imagen y el límite de la función en ese punto y que ambas coincidan

```
f[x_] = x * Sin[x];
a = 0;
```

```
f[a]
```

```
0
```

```
Limit[f[x], x → a, Direction → -1]
```

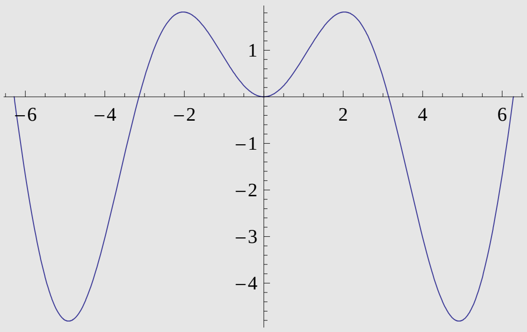
```
0
```

```
Limit[f[x], x → a, Direction → 1]
```

```
0
```

Existe la imagen de la función en x = 0 y existe el límite de la función en x=0 (los límites laterales son iguales) y además son iguales, por tanto la función es continua en x=0

```
Plot[f[x], {x, -2 π, 2 π}]
```



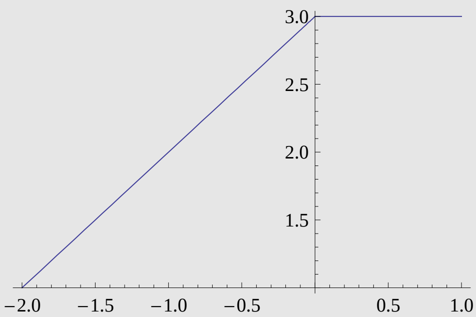
Continuidad de una función en un intervalo

Estudio de la continuidad de una función en un intervalo abierto

Para que una función sea continua en un intervalo abierto, basta con que sea continua en todos los puntos de ese intervalo abierto

```
f[x_] = Which[-2 < x < 0, x + 3, 0 ≤ x < 1, 3];
```

```
Plot[f[x], {x, -2, 1}]
```



Para estudiar la continuidad de la función en el intervalo (-2,1), basta con estudiarla en el único punto posible de discontinuidad, x=0

```
f[0]
```

3

```
Limit[f[x], x → 0, Direction → -1]
```

3

```
Limit[f[x], x → 0, Direction → 1]
```

```
3
```

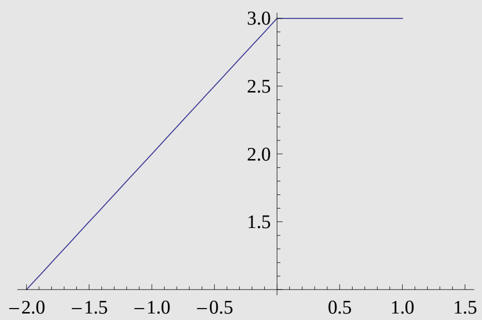
La función es continua en x=0. Por tanto también lo es en el intervalo abierto (-2,1)

Estudio de la continuidad de una función en un intervalo cerrado

Para que una función sea continua en un intervalo cerrado, basta con que sea continua en todos los puntos del intervalo abierto y además que el límite lateral por la derecha de la función en el extremo inferior del intervalo coincida con la imagen de dicho extremo y con que el límite lateral por la izquierda de la función en el extremo superior del intervalo coincida con la imagen de ese extremo

```
f[x_] = Which[-2 ≤ x < 0, x + 3, 0 ≤ x < 1, 3, x == 1, 2];
```

```
Plot[f[x], {x, -2, 1.5}]
```



Para estudiar la continuidad de la función en el intervalo [-2,1], primero con estudiamos la continuidad en el intervalo (-2,1) y según lo visto anteriormente, la función es continua.

Ahora comparamos la imagen de la función en el extremo inferior del intervalo, x = -2, con el límite de la función por la derecha en ese punto

```
f[-2]
```

```
1
```

```
Limit[f[x], x → -2, Direction → -1]
```

```
1
```

Ahora comparamos la imagen de la función en el extremo superior del intervalo, x = 1, con el límite de la función por la izquierda en ese punto

```
f[1]
```

2

```
Limit[f[x], x → 1, Direction → 1]
```

3

No coinciden la imagen de la función con el límite por la izquierda en el punto x=1 . Por tanto la función no es continua en el intervalo cerrado [-2,1]

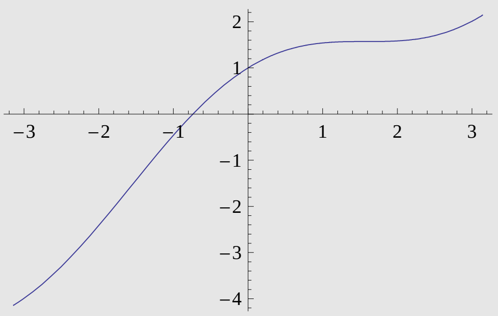
Localización de la raíz de una función

Localización de la raíz de una función utilizando la instrucción FindRoot

⇒ FindRoot[función , {x,x0}]

Encuentra una raíz numérica de la función indicada, a partir del valor x=x0

```
Plot[Sin[x + Pi / 2] + x, {x, -Pi, Pi}]
```




```
FindRoot[Sin[x + Pi / 2] + x, {x, -1}]
```

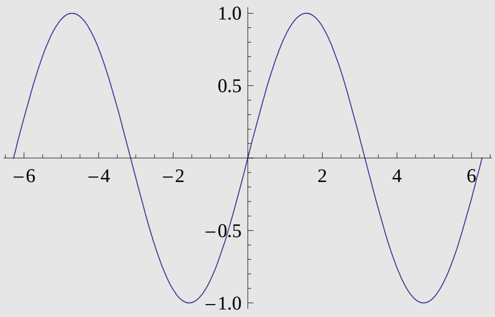
{x → −0.739085}

Si se especifica la opción WorkingPrecision → n, devuelve la raíz con n dígitos significativos

```
FindRoot[Sin[x + Pi / 2] + x, {x, -1}, WorkingPrecision → 20]
```

{x → −0.73908513321516064166}

```
Plot[Sin[x], {x, -2 Pi, 2 Pi}]
```




```
FindRoot[Sin[x], {x, -2}]
```

{x → −3.14159}

```
FindRoot[Sin[x], {x, 0}]
```

{x → 0.}

```
FindRoot[Sin[x], {x, 1}]
```

{x → 0.}

```
FindRoot[Sin[x], {x, 5}]
```

{x → 9.42478}

▼ Localización de la raíz de una función utilizando el Método de Bisección

Localizar la raiz de la ecuación $x^5 + x - 1 = 0$ y calcularla de forma aproximada aplicando el método Bisección las veces necesarias para que el error sea inferior a 10^{-1}.

⇒ 1. Datos iniciales

Definimos la función y la aproximación deseada.

```
f[x_] = x^5 + x - 1;
p = 1;
er = 10^-p;
```

⇒ 2. Localización de la raiz

La función f(x) es continua en un intervalo [a,b]={k,k+1}

a) Si f(a).f(b)<0 la raiz está en el intervalo[a,b]
b) Si f(a).f(b)=0 la raiz es a ó b
c) Si f(a).f(b)>0 hay que buscar otro intervalo.

Iniciamos la búsqueda de la raiz en un intervalo de la forma {a,b}={k,k+1}, empezando con k=-1.

```
k = -1;
```

Coprobamos si se cumple la condición f[a]*f[b]<0. Asimismo, se descartan los casos en que la raíz sea "a" ó "b". Si no hemos localizado la raiz aumentamos el valor de k y repetimos la pregunta

```
{a, b} = {k, k + 1};
If[f[a] * f[b] <= 0, If[f[a] * f[b] == 0,
  If[f[a] == 0, Print["La raiz buscada es a=", a],
   Print["La raiz buscada es b=", b]],
  Print["La raiz buscada está en el intervalo {a,b}=", {a, b}]],
 Print["Buscar otro intervalo;"]]; k = k + 1;
```

```
Buscar otro intervalo;
```

Repetimos este proceso hasta encontrar el intervalo donde se encuentra la raiz buscada.

```
{a, b} = {k, k + 1};
If[f[a] * f[b] <= 0, If[f[a] * f[b] == 0,
  If[f[a] == 0, Print["La raiz buscada es a=", a],
   Print["La raiz buscada es b=", b]],
  Print["La raiz buscada está en el intervalo {a,b}=", {a, b}]],
 Print["Buscar otro intervalo;"]]; k = k + 1;
```

```
La raiz buscada está en el intervalo {a,b}={0, 1}
```

⇒ 3. Cálculo del número de iteraciones

El número "n" de iteraciones necesarias para conseguir una aproximación superior a 10^{-p} , vendrá dada por la función

```
it[p_] = IntegerPart[p * Log[10.] / Log[2.]] + 1
```

```
1 + IntegerPart[3.32193 p]
```

```
A1 = {{"eror<10^(-p)", "Iteraciones"}};
B1 = Table[{10.^-p, it[p]}, {p, 1, 10}];
Grid[Flatten[{A1, B1}, 1], Frame → All]
```

eror<10^(-p)	Iteraciones
0.1	4
0.01	7
0.001	10
0.0001	14
0.00001	17
$1.\times 10^{-6}$	20
$1.\times 10^{-7}$	24
$1.\times 10^{-8}$	27
$1.\times 10^{-9}$	30
$1.\times 10^{-10}$	34

Para conseguir una aproximación inferior a 0.1 serán necesarias 4 iteraciones. y para una aproximación menor que 0.001 serán necesarias 10 iteraciones.

⇒ 4. Primera aproximación y error

Tomamos como primera aproximación el punto medio del intervalo [a,b]

```
n = 1;
x[n] = (a + b) / 2.;
e[n] = (b - a) / 2.;
```

Test de parada

a) Si f(x1)=0, habremos dado con la solución del problema.

ó

b) El error e1< er habremos conseguido la aproximación deseada

```
If[f[x[n]] == 0,
 Print["Parar; La raiz buscada es {n,x[n],f[x[n]],e[n]}=",
  {n, x[n], f[x[n]], e[n]}], If[e[n] < er,
  Print["Parar; La raiz aproximada es {x[n],f[x[n]],e[n]}=",
   {x[n], f[x[n]], e[n]}], Print["Mejorar la Aproximación"]]]
```

```
Mejorar la Aproximación
```

⇒ 5. Segunda aproximación y error

Tomamos un nuevo intervalo

```
If[f[a] * f[x[n]] < 0,
  Print["El nuevo intervalo es {a,b}=", {a, x[n]}];
  {a, b} = {a, x[n]}, Print["El nuevo intervalo es {a,b}=",
   {x[n], b}]; {a, b} = {x[n], b}];
```

```
El nuevo intervalo es {a,b}={0.5, 1}
```

2ª Aproximación y error

```
n = n + 1;
x[n] = (a + b) / 2.;
e[n] = (b - a) / 2.;
```

Test de parada

```
If[f[x[n]] == 0,
 Print["Parar;la raiz buscada es {n,x[n],f[x[n]],e[n]}=",
  {n, x[n], f[x[n]], e[n]}],
 If[e[n] < er, Print[
   "Parar; La raiz aproximada es {n,x[n],f[x[n]],e[n]}=",
   {n, x[n], f[x[n]], e[n]}, " en ", n, " iteraciones"],
  Print["La nueva aproximación es {x[n],e[n]}=", {x[n], e[n]}];
  Print["Mejorar la Aproximación; nº de iteraciones = ", n]]]
```

```
La nueva aproximación es {x[n],e[n]}={0.75, 0.25}
```

```
Mejorar la Aproximación; nº de iteraciones = 2
```

⇒ 6. Mejora de la solución

3ª Aproximación.- Repetimos este grupo de instrucciones hasta alcanzar la aproximación deseada

```
If[f[a] * f[x[n]] < 0,
  Print["El nuevo intervalo es {a,b}=", {a, x[n]}];
  {a, b} = {a, x[n]}, Print["El nuevo intervalo es {a,b}=",
   {x[n], b}]; {a, b} = {x[n], b}];
If[f[a] * f[x[n]] < 0, {a, b} = {a, x[n]}, {a, b} = {x[n], b}];
n = n + 1;
x[n] = (a + b) / 2.;
e[n] = (b - a) / 2.;
If[f[x[n]] == 0,
 Print["Parar;la raiz buscada es {n,x[n],f[x[n]],e[n]}=",
  {n, x[n], f[x[n]], e[n]}],
 If[e[n] < er, Print[
   "Parar; La raiz aproximada es {n,x[n],f[x[n]],e[n]}=",
   {n, x[n], f[x[n]], e[n]}, " en ", n, " iteraciones"],
  Print["La nueva aproximación es {x[n],e[n]}=", {x[n], e[n]}];
  Print["Mejorar la Aproximación; nº de iteraciones = ", n]]]
```

```
El nuevo intervalo es {a,b}={0.75, 1}
```

```
La nueva aproximación es {x[n],e[n]}={0.875, 0.125}
```

```
Mejorar la Aproximación; nº de iteraciones = 3
```

4ª Aproximación.- Repetimos este grupo de instrucciones hasta alcanzar la aproximación deseada

```
If[f[a] * f[x[n]] < 0,
  Print["El nuevo intervalo es {a,b}=", {a, x[n]}];
  {a, b} = {a, x[n]}, Print["El nuevo intervalo es {a,b}=",
   {x[n], b}]; {a, b} = {x[n], b}];
If[f[a] * f[x[n]] < 0, {a, b} = {a, x[n]}, {a, b} = {x[n], b}];
n = n + 1;
x[n] = (a + b) / 2.;
e[n] = (b - a) / 2.;
If[f[x[n]] == 0,
 Print["Parar;la raiz buscada es {n,x[n],f[x[n]],e[n]}=",
  {n, x[n], f[x[n]], e[n]}],
 If[e[n] < er, Print["Parar; La raiz aproximada es {x[n],e[n]}=",
   {x[n], e[n]}, " en ", n, " iteraciones"],
  Print["La nueva aproximación es {x[n],e[n]}=", {x[n], e[n]}];
  Print["Mejorar la Aproximación; nº de iteraciones = ", n]]]
```

```
El nuevo intervalo es {a,b}={0.75, 0.875}
```

```
Parar; La raiz aproximada es {x[n],e[n]}={0.8125, 0.0625} en 4 iteraciones
```