



Explainix

Explain Nix syntax visually. Snippet below showcases most of the language features Nix has to offer.

Hover over any element to highlight its bounds. Click on any element to display help.

Data types

```
{  
  "list of numbers" = [ 123 456.78 ];  
  booleans = [ true false ];  
  null = null;  
  string = "Hello world!";  
  multiline_string = ''  
    This string  
    can span multiple lines.  
  '';  
  uri = https://zaynetro.com/explainix;  
  paths = {  
    relative = ./config/hello.txt;  
    absolute = /var/lib/nginx.log;  
    home_path = ~/Downloads;  
    lookup_path = <nixpkgs>;  
  };  
  "attribute set" = { a = 123; };  
}
```

Language constructs

```
{  
  "recursive set" = rec {  
    x = y;  
    y = 123;  
  };  
  "conditionals" = if x < y then "this" else "otherwise";  
  "let-expressions" =  
    let  
      a = 23;  
      b = 45;
```

```

    in a + b
;
"inheritng attributes" =
  let
    x = 123;
  in {
    inherit x;
    "y" = 456;
  }
;
"functions" =
  let
    concat = x: y: x + y;
  in concat x y
;
"assertions" = assert x != 0; "x is not zero";
"with-expressions" =
  let
    as = { x = "foo"; y = "bar"; };
  in with as; x + y
;
}

```

Operators

```

[
  (2 + 3)
  ("foo" + "bar")
  ([ 1 2 ] ++ [ 3 4 ])
  ({ x = 1; } // { y = 2; })
  (x -> y)
  ({ y = 2; } ? "x")
  ({ y = 1; }.x or 0)
]

```

