

A tour of Nix

24 / 35 Typing system

[prev](#)[next](#)

The Nix language uses dynamic typing and there are `builtin` functions to check the type of a `binding`.

Note: use these functions: `isBool`, `isInt`, `isString`, `isNull`, `isList`, `isAttrs` and `isFunction`.

Do this:

- go through `ex00`, `ex01`, ... and replace X by `isBool` in respect to the `type`
- fix `ex04`, what is the problem?

Note: `()` can either be a `function` or indicates `precedence`.

See also <https://nixos.org/manual/nix/stable/language/values>.

Note: See video [@youtube](#)

```
1 with import <nixpkgs> {};  
2 with lib;  
3 {  
4   ex00 = isAttrs {};  
5   #ex01 = isX "a";  
6   #ex02 = isX (-3);  
7   #ex03 = isX (x: x);  
8   #ex04 = isX (x:x);  
9   #ex05 = isX ("x");  
10  #ex06 = isX null;  
11  #ex07 = isX (y: y+1);  
12  #ex08 = isX [{z}: z] (x: x)];  
13  #ex09 = isX {a=[]};  
14  #ex10 = isX -10; # oh, what is that?  
15 }  
16
```

[reset](#)[solution](#)[run](#)

```
with import <nixpkgs> {};  
with lib;  
{  
  ex00 = isAttrs {};  
  ex01 = isString "a";  
  ex02 = isInt (-3);  
  ex03 = isFunction (x: x);  
  ex04 = isString (x:x); # this is because of url parsing: foo = http://bar.com;  
  ex05 = isString ("x");  
  ex06 = isNull null;  
  ex07 = isFunction (y: y+1);  
  ex08 = isList [{z}: z] (x: x);  
  ex09 = isAttrs {a=[]};  
  ex10 = isInt (-10); # () were missing  
}
```