A tour of Nix

18 / 35 Attribute sets: merging

prev next

Since programming in Nix is all about `attribute sets` it is important to know how to `merge` these using the `//` operator.

```
l = {a="A"; b="B";} // {a="aaa"};
```

will evaluate to:

```
l = {a="aaa"; b="B";};
```

as the later `set` overwrites the attributes from the earlier one.

Now:

- Every exercise `ex00, ex01, ...` should evaluate to what it is compared to, just see the output after hitting 'run' once.

Note: See video @youtube

```
let
  x = { a="bananas"; b= "pineapples"; };
  y = { a="kakis"; c ="grapes"; };
  z = { a="raspberrys"; c= "oranges"; };

  func = {a, b, c ? "another secret ingredient"}: "A drink with: " +
    a + ", " + b + " and " + c;
in
rec {
  ex00=func ( x );
  # hit 'run', you need the output to solve this!
  #ex01=func (y // X );
  #ex02=func (x // { X="lychees";});
  #ex03=func (X // x // z);
}

```

reset solution run

```
with import <nixpkgs> { };
let
  x = { a="bananas"; b= "pineapples"; };
  y = { a="kakis"; c ="grapes"; };
  z = { a="raspberrys"; c= "oranges"; };

  func = {a, b, c ? "another secret ingredient"}: "A drink with: " +
    a + ", " + b + " and " + c;
in
rec {
  ex00=func (x);
  ex01=func (y // x );
  ex02=func (x // { c="lychees";});
  ex03=func (z // x // z);
}
```