The Design Recipe

For the word problems below, assume `animalA` and `animalB` are defined as the data rows for Felix and Midnight, respectively.

Directions : Define a function called `lookup-fixed` , which looks up whether or not an animal is fixed.

Contract and Purpose Statement □

Every contract has three parts...

lookup-fixed:: (r :: Row) -> Boolean

function name *domain* *range*

Consumes an animal, and looks up the value in the fixed column.

what does the function do?

Examples □

Write some examples, then circle and label what changes...

```
examples:
  lookup-fixed ( "animalA" ) is animalA["fixed"]
```

function name *input(s)* *what the function produces*

```
  lookup-fixed ( "animalB" ) is animalB["fixed"]
```

function name *input(s)* *what the function produces*

```
end
```

Definition □

Write the definition, giving variable names to all your input values...

```
fun lookup-fixed( r ):
```

function name *variable(s)*

```
  r["fixed"]
```

what the function does with those variable(s)

```
end
```

Directions : Define a function called `lookup-sex` , which consumes a Row of the animals table and looks up the sex of that animal.

Contract and Purpose Statement □

Every contract has three parts...

lookup-sex:: (r :: Row) -> String

function name *domain* *range*

Consumes an animal, and looks up the sex

what does the function do?

Examples □

Write some examples, then circle and label what changes...

```
examples:
  lookup-sex ( "animalA" ) is animalA["sex"]
```

function name *input(s)* *what the function produces*

```
  lookup-sex ( "animalB" ) is animalB["sex"]
```

function name *input(s)* *what the function produces*

```
end
```

Definition □

Write the definition, giving variable names to all your input values...

```
fun lookup-sex( r ):
```

function name *variable(s)*

```
  r["sex"]
```

what the function does with those variable(s)

```
end
```