

(if cond then else)

↑
what
goes
here?

Number

Boolean

Any

"" - falsey
"_" - truey

JS

?
J
weird

Python

```

data Value:
  | v-num (n :: Number)
  | v-bool (b :: Boolean)
end

```

```

data AST:
  | e-bool (b :: Boolean)
  | e-if (cond :: AST,
         thn :: AST,
         els :: AST)
  | e-num | e-plus
end

```

```

  | e-plus (l, r) =>
    vl = interp(l)
    vr = interp(r)
    => if is-v-bool(vl) or is-v-bool(vr) : false(notnum(vl))

```

```

fun interp(a :: AST) -> Value:
  cases (AST) a:
    | e-num(n) => v-num(n)
    | e-bool(b) => v-bool(b)
    | e-if(cond, thn, els) =>
      vc = interp(cond)
      cases (Value) vc:
        | v-num(n) =>
          -> v-bool(not-bool(vc))
        | v-bool(b) =>
          if b:
            interp(thn)
          else:
            interp(els)
      end
    end
end

```

```

    => if is-v-bool(vl) or is-v-bool(vr) : false(notnum(vl))

```

```
else: vl.n + vr.n  
end
```

```
data IError:
```

```
| not-bool (v :: Value)
```

```
| not-num (vl :: Value, a :: AST, vr :: Value)
```

```
end
```

```
check:
```

```
interp(" (+ true 5) ") raises "not-num"
```

raises-satisfies

```
lam (e :: IError) : e.vl == v-bool(true) end
```

DESUGARING:

(if: cond then els.)

UserAST

[(if: cond then
elseif: cond then
else: els)

desugar :: UserAST → InterpAST

data UserAST

| e-if-n (

conds :: List <UAST>

thens :: List <UAST>

els :: UAST

)

(if cond₁ then₁

(if cond₂ then₂

...
els))

[x + 1 for x in range(100)]

```
lst = []  
for x in range(100):  
    lst.append(x+1)
```

```
lst = []  
iter = range(100).__iter__()  
while True:  
    try:  
        x = iter.next()  
        lst.append(x+1)  
    except StopIteration:  
        break
```

lst

```
for map(i from range(0, 100)):  
    i+1  
end
```

→

```
map(lambda i: i+1, range(0, 100))
```

~

for fold(acc from 0, i from range(0,100)):

acc + i

end

fold(lam(acc, i): acc + i end, 0, range(0,100))