Video 02 REPRESENTING STRUCTURED DATA with SIMON THOMPSON Professor of Logic & Computation, University of Kent



Strings?

```
(2+(3*4)) could be represented as the string (list) "(2+(3*4))" ...
```

... why is that a bad idea?

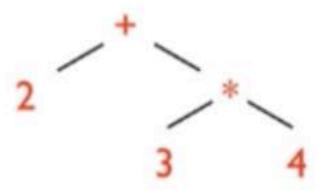


How do we represent them in a program?

Because when we read (2+(3*4)) we see a structure:

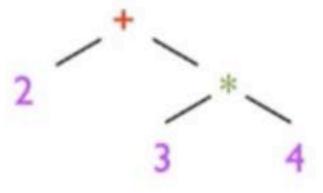
... it's the addition of 2 and (3*4),

... and (3*4) is itself the multiplication of 3 and 4.





How do we represent them in a program?



{add, {num, 2}, {mul, {num, 3}, {num, 4}}}



Defining a type for expressions, expr()

```
-type expr() :: {'num',integer()}

| {'var',atom()}

| {'add',expr(),expr()}

| {'mul',expr(),expr()}.
```



Converting from string() to expr()

```
Going from
```

```
"(2+(3*4))"

to

{add, {num, 2}, {mul, {num, 3}, {num, 4}}}

is called parsing.
```



Converting from expr() to string()

Going from

```
{add, {num, 2}, {mul, {num, 3}, {num, 4}}}

to

"(2+(3*4))"

is called (pretty) printing.
```

