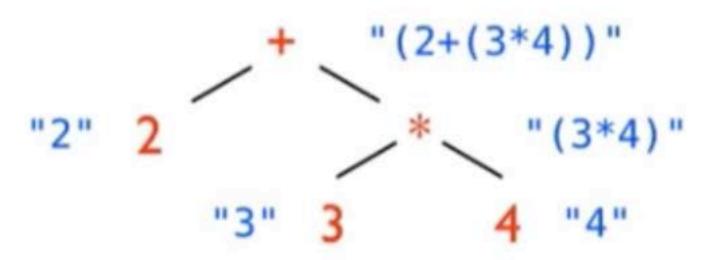


Video 03 USING RECURSION; PRETTY PRINTING with SIMON THOMPSON Professor of Logic & Computation, University of Kent

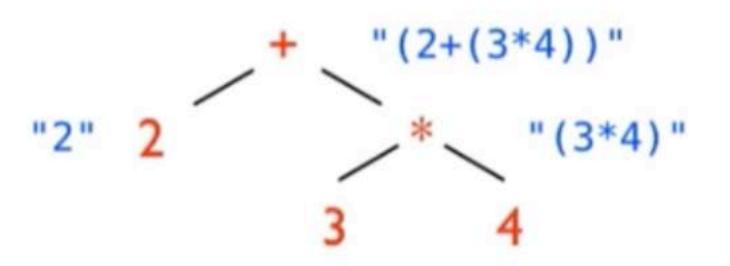


Bottom up ...





Top down ...





Printing

Turn an expression into a string.



Printing

Turn an expression into a string.

```
-spec print(expr()) -> string().

print({num,N}) ->
    integer_to_list(N);

print({var,A}) ->
    atom_to_list(A);

print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";

print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".

print({add,{num,2},{mul,{num,3},{num,4}})
    -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
```



```
print({num,N}) ->
    integer_to_list(N);

print({var,A}) ->
    atom_to_list(A);

print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";

print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".

print({add,{num,2},{mul,{num,3},{num,4}}})
    -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".

print({add,{num,2},{mul,{num,3},{num,4}}})
-> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
print({add, {num, 2}, {mul, {num, 3}, {num, 4}}})
 -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
 -> "("++"2"++"+"+" ("++ print({num,3}) ++"""++ print({num,4}) ++")"++")"
```



```
print({num, N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
print({add, {num, 2}, {mul, {num, 3}, {num, 4}}})
 -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
 -> "("++"2"++"+"+" ("++ print({num,3}) ++"""++ print({num,4}) ++")"++")"
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul, E1, E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
print({add, {num, 2}, {mul, {num, 3}, {num, 4}}})
 -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
 -> "("++"2"++"+"+" ("++ print({num,3}) ++"""++ print({num,4}) ++")"++")"
 -> "("++"2"++"+"+"("++"3"++"*"++"4"++")"++")"
```



```
print({num,N}) ->
    integer_to_list(N);
print({var,A}) ->
    atom_to_list(A);
print({add,E1,E2}) ->
    "(" ++ print(E1) ++ "+" ++ print(E2) ++ ")";
print({mul,E1,E2}) ->
    "(" ++ print(E1) ++ "*" ++ print(E2) ++ ")".
print({add, {num, 2}, {mul, {num, 3}, {num, 4}}})
 -> "("++ print({num,2}) ++"+"++ print({mul,{num,3},{num,4}}) ++")"
 -> "("++"2"++"+"+" ("++ print({num,3}) ++"""++ print({num,4}) ++")"++")"
 -> "("++"2"++"+"+"("++"3"++"*"++"4"++")"++")"
 -> "(2+(3*4))"
```



www.kent.ac.uk/elearning

