20. Expressions and Strings

2015-06-09

The chapters numbered 20, 21,... are of a more fundamental nature. Here the sturcture of *Mathematica* expressions and strings is investigated and commands for working on them are described. *Mathematica* handles many different kinds of things: mathematical fornulas, lists, text, etc. *Mathematica* represents all of these things in one uniform way. But there are two types of objects to be treated: (Mathematical) expressions and strings: **Expressions** are treated in the kernel according to rules stored in the kernel; these rules are either provided (automatically) by the software or by rules prepared during the session by the user.
Strings are text, which is stored as such and is not automtically reworked. But there also special commands to work on strings.

```
$Version
```

```
10.0 for Mac OS X x86 (64-bit) (December 4, 2014)
```

20.1 The Structure of Expressions

A prototypical example of an expression is **`f[x,y]`**. this might be used to represent a mathematical
function f(x, y). This function is named **f**, its tow arguments as **`x`** and **`y`**.
`x + y` is also an expression. When you type it in, *Mathematica* converts it to the standard form: **`Plus[x,y].`** When you print it out, this is displayed as: **`x + y`**.

This applies alos to other "operators" such as ^ (Power) or / (Divide). In fact, everything you type into *Mathematica* and send to the kernel is treated as an expression, except it is tagged as a string by
quotation marks.

`x + y + z`	Plus[x, y, z]
`x y z`	Times[x, y, z]
`x ^ n`	Power[x, n]
`{a, b, c}`	List[a, b, c]
`a -> b`	Rule[a, b]
`a = b`	Set[a, b]
`FullForm[`*`expr`*`]`	give the full form of any expression
`Treeform[`*`expr`*`]`	displays ***expr*** as a tree with different levels at different depths
`Head[`*`expr`*`]`	give the head of an expression

`Input`	`Output / Representation`
`x + y + z`	`x + y + z`
`FullForm[%]`	`Plus[x, y, z]`
`x y`	`x y`
`FullForm[%]`	`Times[x, y]`

Note that subtraction, division respectively are not represented by these operators but by addition, multiplication respectively:

Input	Output
x - y	x - y
FullForm[%]	Plus[x, Times[-1, y]]
x/y	$\frac{x}{y}$
FullForm[%]	Times[x, Power[y, -1]
a + b^c	a + b^c
FullForm[%]	Plus[a, Power[b, c]]
1 + x^2 + (y + z)^2	$1 + x^2 + (y + z)^2$
FullForm[%]	Plus[1, Power[x, 2], Power[Plus[y, z], 2]]

$(1 + x^2 + (y + z)^2)$ "10.0 for Mac OS X x86 (64-bit) (December 4, 2014)"

FullForm[Sqrt[a + x]]

Power[Plus[a, x], Rational[1, 2]]

Input	Output
f[x, y]	f[x, y]
FullForm[f[x,y]]	f[x, y]
FullForm[I]	Complex[0, 1]

FullForm[a + b I]

Plus[a, Times[Complex[0, 1], b]]

The object **f** in an expression **f[x, y, ...]** is called the head of the expression. It is extracted by the command: "**Head**"

Input	Output / Representation
Head[a + b + c]	Plus
Head[{a, b, c}]	List
Head[23456]	Integer
Head[3.14]	Real
Head[Pi]	Symbol
Head[N[Pi]]	Real
Head[f[x, y]]	f
FullForm[x -> Pi]	Rule[x, Pi]
FullForm[Real[x] :> x]	RuleDelayed[Real[x], x]
FullForm[a = b]	b
FullForm[a := c]	Null
a == b	a == b
FullForm[%]	Equal[a, b]
a != b	True
FullForm[%]	True

Hold[a != b]

Hold[a ≠ b]

Input	Output
Clear[a,b]; Remove[a,b] **a != b**	True
BesselJ[1, 5.3]	-0.345961
FullForm[BesselJ[1, 5.3]]	-0.34596083380118675`

```
Head[ BesselJ[1, 5.3] ]          Real
FullForm[ BesselJ[1, 5]]         BesselJ[1, 5]
Head[ BesselJ[1, 5] ]            BesselJ
FullForm[ Real[x] ^=  x ]        x
FullForm[ f[x_] = Sin[a x] ]     Sin[Times[a, x]]
FullForm[ Clear[a] ]             Null
FullForm[ g[x_,c_] := Sin[c x]]  Null
FullForm[ f[x] /. x -> Pi]       f[Pi]
FullForm[  D[f[x], x]  ]         Derivative[1][f][x]
FullForm[ D[f[x], {x,3}] ]       Derivative[3][f][x]
```

```
FullForm[ D[f[x,y,z], {x,7}, {y, 3}, {z,2}] ]
```

Derivative[7, 3, 2][f][x, y, z]

```
D[ Sin[a x] Exp[b y], {x,3}, {y,2} ]
```

$-a^3\, b^2\, e^{b\, y}\, \mathrm{Cos}[a\, x]$

```
FullForm[ D[ Sin[a x] Exp[b y], {x,3}, {y,2} ]  ]
```

Times[-1, Power[a, 3], Power[b, 2], Power[E, Times[b, y]], Cos[Times[a, x]]]

```
FullForm[ Expand[ (x + y)^2 ] ]
```

Plus[Power[x, 2], Times[2, x, y], Power[y, 2]]

```
FullForm[ Series[ Sin[x], {x, 0, 3}] ]
```

SeriesData[x, 0, List[1, 0, Rational[-1, 6]], 1, 4, 1]

```
g[x_, c_] := Sin[c x]
c = 0.3;
FullForm[ Plot[g[x, c], {x, 0, 2 Pi}, PlotPoints -> 7,
PlotRange -> {0, 1}] ]
```

```
Graphics[List[List[List[], List[], List[Directive[Opacity[1.`],
     RGBColor[0.368417`, 0.506779`, 0.709798`], AbsoluteThickness[1.6`]],
    Line[List[List[1.0471975511965976`*^-6, 3.141592653589741`*^-7],
      List[0.01573851326476576`, 0.004721536436458828`],
      List[0.031475979331980326`, 0.009442653470302783`],
      List[0.06295091146640945`, 0.01888415087662969`],
      List[0.1259007757352677`, 0.03776125291872628`],
      List[0.2518005042729842`, 0.07546832913407198`],
      List[0.5035999613484172`, 0.15050590641447326`],
      List[1.0071988754992833`, 0.29758272041708606`],
      List[1.024260092086788`, 0.30246528386704`], List[1.0413213086742927`,
       0.30733992345161537`], List[1.075443741849302`, 0.31706492042007417`],
      List[1.1436886081993207`, 0.33641422332140153`],
      List[1.2801783408993583`, 0.37468175604179915`],
      List[1.5531578062994331`, 0.4492693547101138`],
      List[1.570219022886938`, 0.4538361784670252`],
      List[1.5872802394744425`, 0.45839111280397987`],
      List[1.621402672649452`, 0.4674648362191772`],
      List[1.6896475389994707`, 0.48546438081481735`],
      List[1.8261372716995081`, 0.5208455666679683`],
      List[2.099116737099583`, 0.5889306270455461`],
      List[2.115047335764884`, 0.5927863411330998`],
      List[2.1309779344301845`, 0.5966285156758285`],
      List[2.1628391317607862`, 0.6042718954090972`],
      List[2.22656152642199`, 0.619392339984789`],
      List[2.354006315744397`, 0.6489487381616262`],
      List[2.60889589438921`, 0.7051741759838897`],
      List[2.624826493054511`, 0.708554711209589`],
      List[2.640757091719812`, 0.7119190626813662`],
```

```
List[2.6726182890504138`, 0.7185989073604842`],
List[2.7363406837116173`, 0.7317610355382862`],
List[2.863785473034024`, 0.7572783402640862`],
List[3.1186750516788377`, 0.8049567168538073`],
List[3.1342931552360995`, 0.8077278943634676`],
List[3.1499112587933613`, 0.8104813396416324`],
List[3.1811474659078844`, 0.8159347921060768`],
List[3.243619880136931`, 0.8266262768331999`],
List[3.3685647085950245`, 0.8471345555019837`],
List[3.3841828121522863`, 0.8496149847690236`],
List[3.399800915709548`, 0.852076762247884`],
List[3.4310371228240713`, 0.8569441460759116`],
List[3.493509537053118`, 0.8664527419193523`],
List[3.6184543655112114`, 0.8845536970389555`],
List[3.634072469068473`, 0.886729444489882`],
List[3.649690572625735`, 0.888885725370577`],
List[3.680926779740258`, 0.8931396985003961`],
List[3.7433991939693048`, 0.9014119918350367`],
List[3.868344022427398`, 0.9170039430262583`],
List[3.88396212598466`, 0.9188627866084563`],
List[3.8995802295419217`, 0.9207014581899615`],
List[3.930816436656445`, 0.9243181243355426`],
List[3.9932888508854916`, 0.9313076463470268`],
List[4.118233679343585`, 0.9443030073195783`],
List[4.135175533421137`, 0.9459633569266468`],
List[4.152117387498688`, 0.9475992701035337`],
List[4.186001095653792`, 0.9507976187613603`],
List[4.253768511963999`, 0.9568992619262766`],
List[4.270710366041551`, 0.9583629723131539`],
List[4.287652220119103`, 0.9598019259589905`],
List[4.321535928274207`, 0.9626054149819695`],
List[4.389303344584413`, 0.9679137195536496`],
List[4.406245198661965`, 0.9691783711420575`],
List[4.423187052739516`, 0.9704179866025578`],
List[4.45707076089462`, 0.9728219816988212`],
List[4.524838177204828`, 0.9773281728140472`],
List[4.5417800312823795`, 0.9783916750787421`],
List[4.558721885359931`, 0.9794299032144943`],
List[4.592605593515035`, 0.9814304304733779`],
List[4.660373009825242`, 0.9851270591980997`],
List[4.677314863902794`, 0.9859876541226044`],
List[4.694256717980345`, 0.9868227786963885`],
List[4.7281404261354485`, 0.9884165311574825`],
List[4.745082280213`, 0.9891751178744654`],
List[4.762024134290552`, 0.9899081519011373`],
List[4.795907842445656`, 0.9912974868002379`],
List[4.812849696523208`, 0.9919537517829199`],
List[4.82979155060076`, 0.9925843922967245`],
List[4.863675258755864`, 0.9937687354164266`],
List[4.8806171128334155`, 0.994322407427991`],
List[4.897558966910967`, 0.9948503937828024`],
List[4.93144267506607`, 0.9958292556295841`],
List[4.948384529143622`, 0.9962801058352799`],
List[4.965326383221173`, 0.9967052198123268`],
List[4.982268237298725`, 0.9971045865790436`],
List[4.999210091376277`, 0.9974781958188591`],
List[5.016151945453829`, 0.9978260378805793`],
List[5.033093799531381`, 0.9981481037786359`],
List[5.050035653608933`, 0.9984443851933189`],
List[5.0669775076864845`, 0.9987148744709914`],
List[5.083919361764036`, 0.9989595646242871`],
List[5.100861215841588`, 0.9991784493322913`],
List[5.11780306991914`, 0.9993715229407035`],
List[5.134744923996692`, 0.9995387804619843`],
List[5.151686778074244`, 0.9996802175754833`],
List[5.168628632151796`, 0.9997958306275516`],
List[5.185570486229347`, 0.9998856166316354`],
List[5.202512340306899`, 0.999949573268354`],
```

```
      List[5.2193978390518225`, 0.999987614885095`],
      List[5.236283337796747`, 0.9999999960684126`],
      List[5.253168836541671`, 0.9999867165005963`],
      List[5.270054335286595`, 0.9999477765224097`],
      List[5.286939834031519`, 0.9998831771330821`],
      List[5.303825332776443`, 0.9997929199902821`],
      List[5.3207108315213665`, 0.9996770074100759`],
      List[5.33759633026629`, 0.999535442366875`],
      List[5.354481829011214`, 0.9993682284933221`],
      List[5.371367327756138`, 0.9991753700802734`],
      List[5.388252826501063`, 0.9989568720766131`],
      List[5.405138325245987`, 0.9987127400891641`],
      List[5.4220238239909l`, 0.9984429803825368`],
      List[5.438909322735835`, 0.9981475998789676`],
      List[5.472680320225683`, 0.997480007457001`],
      List[5.489565818970607`, 0.9971078126695266`],
      List[5.5064513177155305`, 0.9967100313465171`],
      List[5.540222315205379`, 0.9958377505796705`],
      List[5.557107813950303`, 0.9953632735192133`],
      List[5.573993312695228`, 0.994863254689406`],
      List[5.607764310185075`, 0.9937866437002683`],
      List[5.624649808929999`, 0.9932100791675845`],
      List[5.641535307674923`, 0.9926080281181355`],
      List[5.675306305164771`, 0.9913275289188767`],
      List[5.742848300144467`, 0.9884614158468334`],
      List[5.759733798889391`, 0.9876814283264425`],
      List[5.776619297634316`, 0.9868760961586199`],
      List[5.810390295124163`, 0.9851894811922265`],
      List[5.877932290103859`, 0.9815130682767864`],
      List[5.8948177888487825`, 0.9805309385603335`],
      List[5.911703287593706`, 0.9795236476833881`],
      List[5.945474285083555`, 0.9774336864843733`],
      List[6.013016280063251`, 0.9729530106412861`],
      List[6.029901778808175`, 0.9717703514514141`],
      List[6.046787277553099`, 0.970562755904296l`],
      List[6.080558275042947`, 0.9680728803286482`],
      List[6.148100270022644`, 0.9627952991271506`],
      List[6.1649857687675675`, 0.9614140524693018`],
      List[6.181871267512491`, 0.9600081352046189`],
      List[6.215642265002339`, 0.9571224337944643`],
      List[6.232527763747263`, 0.9556427236982585`],
      List[6.249413262492187`, 0.9541384910918506`],
      List[6.2662987612371115`, 0.9526097745749792`],
      List[6.283184259982035`, 0.9510566133756586`]]]]]],
 List[Rule[DisplayFunction, Identity],
  Rule[
   AspectRatio,
   Power[GoldenRatio, -1]], Rule[
   Axes,
   List[True, True]], Rule[
   AxesLabel,
   List[None, None]], Rule[
   AxesOrigin,
   List[0, 0]],
  RuleDelayed[DisplayFunction,
   Identity], Rule[
   Frame,
   List[List[False, False], List[False, False]]],
  Rule[FrameLabel, List[List[None, None], List[None, None]]],
  Rule[
   FrameTicks,
   List[List[Automatic, Automatic],
    List[Automatic, Automatic]]],
  Rule[GridLines, List[None, None]],
  Rule[
   GridLinesStyle,
   Directive[GrayLevel[0.5`, 0.4`]]],
  Rule[Method, List[Rule["DefaultBoundaryStyle", Automatic],
```

```
    Rule["ScalingFunctions", None]]],
  Rule[PlotRange, List[List[0.`, 6.283184259982035`], List[0, 1]]],
  Rule[
   PlotRangeClipping,
   True], Rule[
   PlotRangePadding,
   List[List[Scaled[0.02`], Scaled[0.02`]], List[0, 0]]],
  Rule[Ticks, List[Automatic, Automatic]]]]
```

`FullForm[ FindRoot[Sin[x] + Cos[x], {x,1} ]  ]`

```
List[Rule[x, 5.497787143782138`]]
```

`FindRoot[Sin[x] + Cos[x], {x,1} ]`

{x → 5.49779}

`FullForm[ Integrate[x^2/(x^2 + y^2 + z^2), x] ]`

```
Plus[x, Times[-1, Power[Plus[Power[y, 2], Power[z, 2]], Rational[1, 2]],
  ArcTan[Times[x, Power[Plus[Power[y, 2], Power[z, 2]], Rational[-1, 2]]]]]]
```

`Integrate[x^2/(x^2 + y^2 + z^2), x]`

$$x - \sqrt{y^2 + z^2}\ \text{ArcTan}\left[\frac{x}{\sqrt{y^2 + z^2}}\right]$$

`eq = f1 =  a + bx + (c/2) x^2 + d 3.33 y + Pi z^3 == 0`

$a + bx + 0.15\, x^2 + 3.33\, d\, y + \pi\, z^3 == 0$

`FullForm[%]`

```
Equal[Plus[a, bx, Times[0.15`, Power[x, 2]],
  Times[3.33`, d, y], Times[Pi, Power[z, 3]]], 0]
```

`sig[x_] = If[x > 0, 1, 0]//FullForm`

```
If[Greater[x, 0], 1, 0]
```

`Which[x >= 2, 2, x > 0, 1, x <= 0, 0] //FullForm`

```
Which[GreaterEqual[x, 2], 2, Greater[x, 0], 1, LessEqual[x, 0], 0]
```

`DSolve[y'[x] == a y[x], y[x], x]`

{{y[x] → $e^{a\,x}$ C[1]}}

`DSolve[y'[x] == a y[x], y[x], x]//FullForm`

```
List[List[Rule[y[x], Times[Power[E, Times[a, x]], C[1]]]]]
```

`DSolve[y'[x] == a y[x], y, x]//FullForm`

```
List[List[Rule[y, Function[List[x], Times[Power[E, Times[a, x]], C[1]]]]]]
```

`s = NDSolve[{y'[x] == 2 y[x], y[0] == 1}, y, {x,0,3}]`

{{y → InterpolatingFunction[Domain {{0., 3.}} Output scalar]}}

`FullForm[s]`

```
List[List[Rule[y, InterpolatingFunction[List[List[0.`, 3.`]],
    List[5, 7, 1, List[60], List[4], 0, 0, 0, 0, Automatic, List[], List[], False],
    List[List[0.`, 0.00009738090631613134`, 0.0001947618126322667`,
      0.002452790829421534`, 0.004710819846210806`,
      0.006968848863000077`, 0.013917149286260115`, 0.020865449709520152`,
      0.027813750132780188`, 0.0347620505560402`, 0.048658651402560`,
      0.0625552522490803`, 0.0764518530956004`, 0.0903484539421205`,
      0.104245054788640`, 0.1373307538564709`, 0.1704164529243013`,
      0.203502151992131`, 0.2365878510599620`, 0.2696735501277924`,
      0.302759249195622`, 0.3635877096947477`, 0.4132721246220702`,
      0.462956539549392`, 0.512640954476715`, 0.5623253694040379`,
      0.6120097843313604`, 0.661694199258683`, 0.7113786141860056`,
      0.7786152280933989`, 0.8458518420007923`, 0.913088455908185`,
      0.9803250698155792`, 1.0475616837229726`, 1.11479829763036`,
      1.1820349115377595`, 1.249271525445153`, 1.3264166337113794`,
      1.4035617419776059`, 1.4807068502438323`, 1.5578519585100588`,
      1.6349970667762852`, 1.7121421750425116`, 1.789287283308738`,
      1.8664323915749645`, 1.943577499841191`, 2.0207226081074174`,
      2.097867716373644`, 2.1750128246398703`, 2.252157932906097`,
      2.3293030411723232`, 2.4064481494385497`, 2.483593257704776`,
      2.5607383659710026`, 2.637883474237229`, 2.7150285825034555`,
      2.7921736907696824`, 2.869318799035909`, 2.934659399517954`, 3.`]],
    List[Developer`PackedArrayForm, List[0, 2, 4, 6, 8, 10, 12, 14, 16, 18,
      20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54,
      56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90,
      92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120],
     List[1.`, 2.`, 1.0001947997521838`, 2.000389599489592`,
      1.0003896374513108`, 2.0007792748878432`, 1.0049176795605694`,
      2.0098353590171114`, 1.009466216849263`, 2.01893243349`,
      1.0140353420843817`, 2.0280706839592924`, 1.0282253191766433`,
      2.056450636179455`, 1.0426139022151513`, 2.0852278063598693`,
      1.0572038335153966`, 2.114407666848378`, 1.0719979305974832`,
      2.1439958610068506`, 1.1022100871933582`, 2.2044201727313335`,
      1.1332737097416843`, 2.266547420165146`, 1.1652128039089493`,
      2.3304256084842208`, 1.1980520387416158`, 2.3961040773397664`,
      1.2318167829111342`, 2.4636335656718344`, 1.3160851677211143`,
      2.6321703304859287`, 1.4061183129464767`, 2.81223662071581`,
      1.5023106206993637`, 3.0046212475782044`, 1.6050834298805297`,
      3.210166859098292`, 1.714886908654855`, 3.4297738152940798`,
      1.8322020244517963`, 3.664404046896526`, 2.069227778167112`,
      4.13845552606468`, 2.28540739258492`, 4.570814756176935`,
      2.5241721094471226`, 5.0483442489195784`, 2.7878814275030375`,
      5.575762846050637`, 3.0791414029144186`, 6.158282802849896`,
      3.400830355703659`, 6.801660707387107`, 3.7561273064747343`,
      7.512254606492679`, 4.148543404907389`, 8.297086803914043`,
      4.745660015371183`, 9.491320005694217`, 5.428721991663125`,
      10.857443938482533`, 6.210099831274511`, 12.420199677943787`,
      7.103944551983663`, 14.20788911732097`, 8.126443947445066`,
      16.25288788335351`, 9.296115806286672`, 18.59223159872308`,
      10.63414325426862`, 21.268286489534916`, 12.164758389301888`,
      24.329516757146703`, 14.194198633714443`, 28.388397223042464`,
      16.562209316674867`, 33.124418534488996`, 19.32527397458559`,
      38.65054784574909`, 22.549299256880676`, 45.09859838931502`,
      26.31118697647891`, 52.62237380569923`, 30.700668444835213`,
      61.401336721900584`, 35.82244479501543`, 71.64488939149506`,
      41.79868439608462`, 83.59736856118532`, 48.77193690325199`,
      97.54387353751697`, 56.9085334539228`, 113.81706659326585`,
      66.40255412247885`, 132.80510787832014`, 77.48045725979712`,
      154.96091409170492`, 90.40648114391`, 180.81296178844786`,
      105.48894678591479`, 210.97789298926946`, 123.0876122286883`,
      246.17522377756075`, 143.62225375996462`, 287.2445067267101`,
      167.5826787245598`, 335.16535652358664`, 195.54040876864966`,
      391.0808164573397`, 228.16231219370542`, 456.3246231272927`,
      266.2265105887658`, 532.4530197071877`, 310.6409391578082`,
      621.2818765999723`, 354.00778666736096`, 708.0155723847522`,
      403.4288383751544`, 806.8576766456067`]], List[Automatic]]]]]
```


The output of **NDSolve[]** contains a large amount of data. This is contained in the file, though it is shown in the print. It is recommended to delete these before storing the file. A file containing the output of several **NDSolve[]**'s for long intervals of integration may well exceed the capacity of a floppy disk.

20.2 Strings

```
sn = 1234567890
```

```
1 234 567 890
```

```
Head[sn]
```

```
Integer
```

```
FullForm[sn]
```

```
1 234 567 890
```

```
Characters[sn]
```

```
Characters[1 234 567 890]
```

Above there are still expressions. A string is tagged by quotation marks:

```
st = "1234567890"
```

```
1234567890
```

```
Head[st]
```

```
String
```

```
FullForm[st]
```

```
"1234567890"
```

```
sv = Characters[st]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 0}
```

```
Position[sv,5]
```

```
{}
```

```
sw = FullForm[sv]
```

```
List["1", "2", "3", "4", "5", "6", "7", "8", "9", "0"]
```

```
s9 = "9"
```

```
9
```

```
Position[sv,s9]
```

```
{{9}}
```

```
lp = Characters[ToString[Pi]]
```

```
{P, i}
```

```
np = N[Pi,17]
```

```
3.1415926535897932
```

```
lp = Characters[ToString[%]]
```

```
{3, ., 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2}
```

```
Position[lp,s9]
```

```
{{7}, {14}, {16}}
```

ToString[*expr*]	transforms ***expr*** to a string
ToExpression[*string*]	transforms ***string*** to an expression
Characters[*string*]	gives a list of the characters in a string
StringJoin[*s1,s2,...*] **StringJoin[{*s1,s2,...*}]**	gives a string consisting of the concatenation of the strings ***s1,s2,...***

```
f = {u[2] - 2 * u[1] + u[0], u[3] - 2 * u[2] + u[1]}
```

```
{u[0] - 2 u[1] + u[2], u[1] - 2 u[2] + u[3]}
```

```
FullForm[f]
```

```
List[Plus[u[0], Times[-2, u[1]], u[2]], Plus[u[1], Times[-2, u[2]], u[3]]]
```

```
u[i_] := ToExpression["u" <> ToString[i]]
```

```
f
```

```
{u0 - 2 u1 + u2, u1 - 2 u2 + u3}
```

```
lx = {"x1", "x2", "x3"}
```

```
{x1, x2, x3}
```

```
FullForm[lx]
```

```
List["x1", "x2", "x3"]
```

```
StringJoin[lx]
```

```
x1x2x3
```

```
Characters[lx]
```

```
{{x, 1}, {x, 2}, {x, 3}}
```

```
StringJoin[%]
```

```
x1x2x3
```

```
D[lx, x1]
```

```
{0, 0, 0}
```

```
D[ToExpression[lx], x1]
```

```
{1, 0, 0}
```

20.3 Special Ways to Input Expressions

Mathematica converts the input **x + y** into **Plus[x, y]**. There are several ways to type the input for an expression:

f[x]	standard form	
f @ x	prefix form for	**f[x]**
x // f	postfix form for	**f[x]**

Input Output / Representation

```
Clear[f]
```

`FullForm[ f[x] ]`	`f[x]`
`FullForm[ f @ x ]`	`f[x]`
`x // f`	`f[x]`

```
FullForm[N[Pi]]
```

3.141592653589793`

```
FullForm[N @ Pi]
```

3.141592653589793`

```
FullForm[Pi // N]
```

3.141592653589793`

`f[x, y]`	standard form
`x ~ f ~ y`	infix form for **f[x, y]**

Input	Output / Representation
`FullForm[ f[x, y] ]`	`f[x, y]`
`FullForm[ x ~ f ~ y ]`	`f[x, y]`
`x + y // f`	`f[x + y]`
`f @ x + y`	`y + f[x]`
`f @ (x + y)`	`f[x + y]`
`x y // f`	`f[x y]`
`f @ x y`	`y f[x]`
`f @ (x y)`	`f[x y]`
`3^(1/4) + Pi //N`	`4.45767`
`N[ 3^(1/4) + Pi ]`	`4.45767`
`N @ 3^(1/4) + Pi`	`4.45767`
`{a, b, c} ~ Join ~ {d, e}`	`{a,b,c,d,e}`

20.3.1 Precedence of the Various Forms of Operations

Mathematica interprets **a + b^c** as **a + (b ^ c)** . The operator ^ has higher precedence than + .

Every special input form is assigned a definite precedence. This includes not only the traditional mathemtical operators, but also forms as ->, := or the semicolons used to separate expressions in a *Mathematica* program.

The precedence of prefix form and postfix form differs: The postfix form `//` has very low precedence.
`// f` at the end of an expression implies that `f` is applied to the whole expression preceeding it.
Loosely speaking, the lower the precedence of an operator, the higher the range on which it operates.

```
Clear[f]
x + y + z//f
```

f[x + y + z]

```
x + y - z  // f
```

f[x + y - z]

The prefic form `@` has much higher precedence: `f @ x + y` is equivalent to **f[x] + y** :

```
f @ x + y
```

```
y + f[x]
```

```
f @ (x + y)
```

```
f[x + y]
```

prefixform	High Precedence	Small Range
postfixform	Low Precedence	Large Range

The forms of the operators and their precedences are listed in Tables A.2.7 of the *Mathemtica* book.
These tables of input forms are arranged in decreasing order of precedence. Input forms in the same box
have the same precedence.

20.4 Part of Expressions

Expressions are a particular kind of lists. One can refer to parts of any expression as one refers to parts of a list (cf. Chap.5).

Part[*expr***, n]** or *expr***[[n]]**	the **n-th** part of *expr*
Part[*expr***, {n1, n2, ... }]** or *expr***[[n1, n2, ...]]**	a combination of parts of an expression
ReplacePart[*expr***,** *elem***, n]**	replace the **n-th** part of *expr* by *elem*

```
Clear[a,b,c]
```

```
{a,b,c}[[2]]
```

```
b
```

```
(a + b + c)[[2]]
```

```
b
```

```
(a + b + c)[[-1]]
```

```
c
```

```
(a + b + c)[[0]]
```

```
Plus
```

```
Head[ (a + b + c) ]
```

```
Plus
```

```
(a b c) [[1]]
```

```
a
```

```
(a b c) [[0]]
```

```
Times
```

```
h = f[g[a], g[b]]
```

```
f[g[a], g[b]]
```

```
FullForm[h]
```

```
f[g[a], g[b]]
```

```
h[[0]]
```

```
f
```

```
h[[1]]
```

```
g[a]
```

```
h[[2]]
```

```
g[b]
```

```
h[[3]]
```

```
f[g[a], g[b]]⟦3⟧
```

Part::"partw" : "
Part 3 of f[g[a],g[b]] does not exist. >>

```
h[[1,1]]
```

```
a
```

```
h[[1,0]]
```

```
g
```

The assignemt of indices to expressions is done on the basis of the internal *Mathematica* form of the expression, as shown by **FullForm.**

```
FullForm[x/y]
```

```
Times[x, Power[y, -1]]
```

```
(x/y)[[1]]
```

```
x
```

```
(x/y)[[1,1]]
```

$\frac{x}{y}$⟦1, 1⟧

```
(x/y)[[2]]
```

$\frac{1}{y}$

```
(x/y)[[2,1]]
```

```
y
```

```
(x/y)[[2,0]]
```

```
Power
```

```
(x/y)[[2,2]]
```

```
-1
```

20.5 Expressions as Trees

The best way to investigate the hierarchical tree structure of a list representing a complicated expression is to use the following commands:

TreeForm[*expr*]	Give the representation tree for ***expr***
Depth[*expr*]	The total number of levels in ***expr***
Position[*list*, *pattern*]	give position of ***pattern*** in ***list***

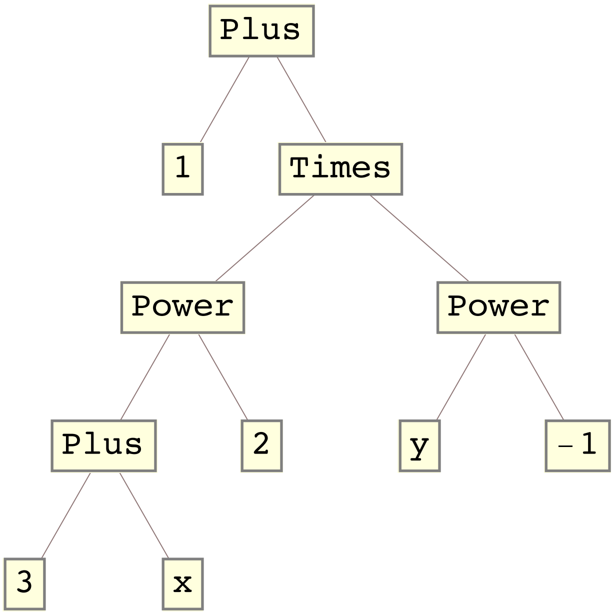
```
t = 1 + (3 + x)^2 / y
```

$$1+\frac{(3+x)^2}{y}$$

```
FullForm[t]
```

```
Plus[1, Times[Power[Plus[3, x], 2], Power[y, -1]]]
```

```
h = TreeForm[t]
```



In this expression the top node of the tree consists of a **Plus**. From this node come two "branches",
x^3 and **(1 + x)^2.** From the **x^3,** there are then two branches, **x** and **3**.

```
Depth[t]
```

5

The very first head (in the example above : **Plus** at the beginning of the first line) is assigned a level 0. Including this into the count of levels gives the depth 5.

```
Position[t, +]
```

```
Position[t,Plus]
```

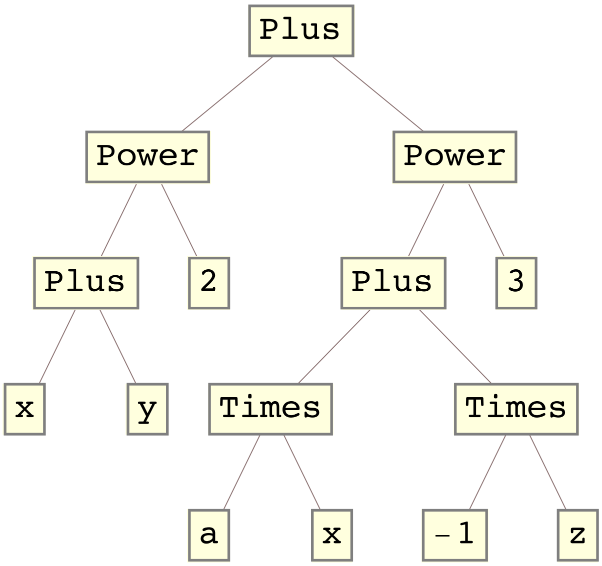
```
{{0}, {2, 1, 1, 0}}
```

Input	Output / Representation
Position[t, Plus]	{{0}, {2, 1, 1, 0}}
Position[t, Power]	{{2, 1, 0}, {2, 2, 0}}
Position[t,1]	{{1}}
Position[t,-1]	{{2, 2, 2}}
Position[t,3]	{{2, 1, 1, 1}}
Position[t,Times]	{{2, 0}}

20.6 Levels in Expressions

Many operators allow one to specify the levels in an expression on which they should act.

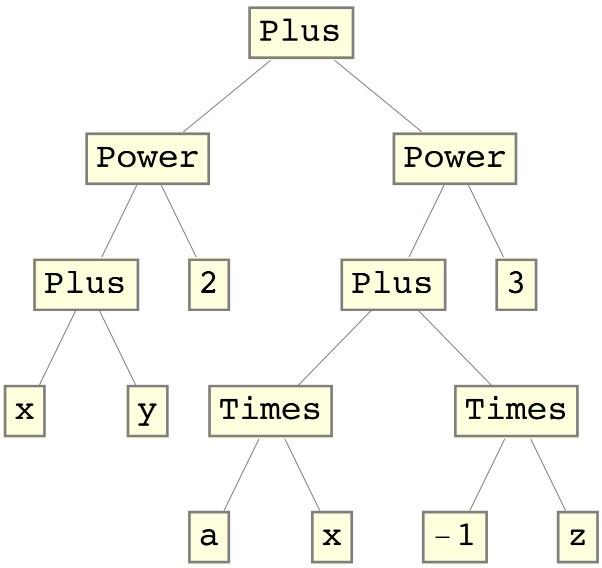
```
h = ( (x + y)^2 + (a x - z)^3 ) // TreeForm
```



Input	Output / Representation
Position[h, x, 1]	{}
Position[h, x, 2]	{}
Position[h, x, 3]	{}
Position[h, x, 4]	{{1, 1, 1, 1}}
Position[h, x, 5]	{{1, 1, 1, 1}, {1, 2, 1, 1, 2}}
Position[h, z]	{{1, 2, 1, 2, 2}}

Position[*expr*,*form*,n]	give the positions at which ***form*** occurs down to level **n**
Position[*expr*,*form*,{n}]	give the positions of ***form*** exactly at level **n**
Level[*expr*, *lev*]	a list of the parts of ***expr*** at the levels specified by ***lev***


```
h = ( (x + y)^2 + (a x - z)^3 ) // TreeForm
```

```
Depth[h]
```

6

```
Position[h, x]
```

{{1, 1, 1, 1}, {1, 2, 1, 1, 2}}

```
Position[h, x, {5}]
```

{{1, 2, 1, 1, 2}}

```
Position[h, Times]
```

{{1, 2, 1, 1, 0}, {1, 2, 1, 2, 0}}

```
Position[h, Power]
```

{{1, 1, 0}, {1, 2, 0}}

```
Position[h, Plus]
```

{{1, 0}, {1, 1, 1, 0}, {1, 2, 1, 0}}

```
Level[h, 1]
```

$\{(x+y)^2 + (a\,x - z)^3\}$

```
Level[h,2]
```

$\{(x+y)^2, (a\,x-z)^3, (x+y)^2 + (a\,x-z)^3\}$

```
Level[h, 5]
```

$\{x, y, x+y, 2, (x+y)^2, a, x, a\,x, -1, z, -z, a\,x-z, 3, (a\,x-z)^3, (x+y)^2 + (a\,x-z)^3\}$

```
Level[h, 6]
```

$\{x, y, x+y, 2, (x+y)^2, a, x, a\,x, -1, z, -z, a\,x-z, 3, (a\,x-z)^3, (x+y)^2 + (a\,x-z)^3\}$

20.7 Changing Parts of Lists Representing Expressions

In a notebook session commands can be changed easily by returning to them with the cursor and editing them. Thus one may avoid to type lengthy expressions in slightly altered form. Such an approach is not possible when working with a text-based interface (cf. sect. 2.1). In such a case the method for changing parts of command lists described here may present some advantage. This way is also useful for procedures and programmes (cf. § **23.2.1**). At first the position of the letter, symbol or strings of symbols is found with the help of the command **Position** so that **ReplacePart** can be targeted precisely. The search for the position can be rendered more specific by given the level of the wanted symbol.

Part[*expr*, n]	**= value**	give that part of *expr* characterized
***expr*[[n]]**	**= value**	by adress **n** the new value **value**
ReplacePart[*expr*, *elem*, n]		replace the **n-th** part of ***expr*** by ***elem***

One can manipulate parts of expressions just as one manipulates parts of lists. The use of assignments as shown below is clumsy and tedious.

```
t = 1 + (3 + x)^2 / y
```

$$1 + \frac{(3+x)^2}{y}$$

```
FullForm[t]
```

Plus[1, Times[Power[Plus[3, x], 2], Power[y, -1]]]

```
t[[2, 1, 1]] = x
```

x

```
t
```

$1 + \frac{x^2}{y}$

```
FullForm[t]
```

Plus[1, Times[Power[x, 2], Power[y, -1]]]

```
t = 1 + (3 + x)^2 / y
```

$1 + \frac{(3 + x)^2}{y}$

```
t[[2,1,1,1]] = 4
```

4

```
t
```

$1 + \frac{(4 + x)^2}{y}$

```
t[[2,1,2]] = z
```

z

```
t
```

$1 + \frac{(4 + x)^z}{y}$

```
t[[2,2,2]]
```

-1

```
t[[2,2,1]]
```

y

```
t[[2,2,1]] = w
```

w

```
t
```

$1 + \frac{(4 + x)^z}{w}$

```
{a, b, c, d, e}[[{2, 4}]]
```

{b, d}

```
h = a + b + c + d + e
```

a + b + c + d + e

```
FullForm[h]
```

Plus[a, b, c, d, e]

```
h[[{2,4}]]
```

b + d

```
h[[2, 4]]
```

(a + b + c + d + e)⟦2, 4⟧

Part::"partd" :
Part specification (a + b + c + d + e)[2,,4] is longer than depth of object. >>

```
ReplacePart[h,aa,5]
```

a + aa + b + c + d

Replacing parts of expressions as shown below in this subsection is clumsy and tetious. An easier method to do this is based on the use of the commands **Position** and **ReplacePart** as described in section 20.7. But before this can be done properly some insight should be gained into the structure of lists representing expressions.

```
a + b + c + d
```

a + b + c + d

```
ReplacePart[%, x^2, 3]
```

$a + b + d + x^2$

```
ReplacePart[a + b + c + d, x^2,3]
```

$a + b + d + x^2$

```
h = ( (x + y)^2 + (a x - z)^3 )
```

$(x + y)^2 + (a x - z)^3$

Now we want to change the symbol **x** in the second term to **w**; thereafter the exponent of the second term from **3** to **4**.

```
Position[h,x]
```

{{1, 1, 1}, {2, 1, 1, 2}}

```
h1 = ReplacePart[h, w, %[[2]] ]
```

$(x + y)^2 + (a w - z)^3$

```
Position[h1, 3]
```

{{2, 2}}

```
ReplacePart[h1, 4, %//Flatten]
```

$(x + y)^2 + (a w - z)^4$

```
t = 1 + (3 + x)^2 / y
```

$1 + \frac{(3 + x)^2}{y}$

```
FullForm[t]
```

Plus[1, Times[Power[Plus[3, x], 2], Power[y, -1]]]

```
ReplacePart[t,w,2]
```

1 + w

```
ReplacePart[t,w,{2,1}]
```

$1 + \frac{w}{y}$

```
ReplacePart[t,w,{2,2}]
```

$1 + w (3 + x)^2$

```
ReplacePart[t,w,{2,2,1}]
```

$1+\frac{(3+x)^2}{w}$

```
ReplacePart[t,w,{2,2,2}]
```

$1+(3+x)^2\, y^w$

```
t = 1 + (3 + x)^2 / y
```

$1+\frac{(3+x)^2}{y}$

```
tt = t /. {y -> w, 3 -> 4, 2 -> 7}
```

$1+\frac{(4+x)^7}{w}$

```
tt /. + -> -
```

Syntax::sntxf: "tt /. + " cannot be followed by "-> -".

ttt = tt /. Plus -> Minus

$\text{Minus}\left[1, \frac{\text{Minus}[4, x]^7}{w}\right]$

Minus::"argx" :
Minus called with 2 arguments; 1 argument is expected. >>

```
ttt
```

$\text{Minus}\left[1, \frac{\text{Minus}[4, x]^7}{w}\right]$

Minus::"argx" :
Minus called with 2 arguments; 1 argument is expected. >>

```
Clear[x]
Minus[x]
```

$-x$

```
FullForm[%]
```

Times[-1, x]

20.8 Taking Parts of Expressions

These list operations can be applied to any kind of expression.

Take[*expr*, n]	give the first **n** elements of ***expr***
Take[*expr*,-n]	give the last **n** elements of ***expr***
Take[*expr*,{m,n}]	**m**-th to **n**-th elements of ***expr***

```
t = 1 + x + x^2 + x^3
```

$1+x+x^2+x^3$

```
Take[t,2]
```

$1+x$

Take[t,-2]

$x^2 + x^3$

Take[t, {2,3}]

$x + x^2$

Take[t, {-1,4}]

x^3

Take[t, {-2,4}]

$x^2 + x^3$

h = ((x + y)^2 + (a x - z)^3)

$(x + y)^2 + (a\,x - z)^3$

Now we want to obtain the content of the second bracket.

po = Position[h, a x - z]

{{2, 1}}

g = ReplacePart[h,1,po]

$1 + (x + y)^2$

The command **Take[]** may also be useful to get parts of a lengthy output.

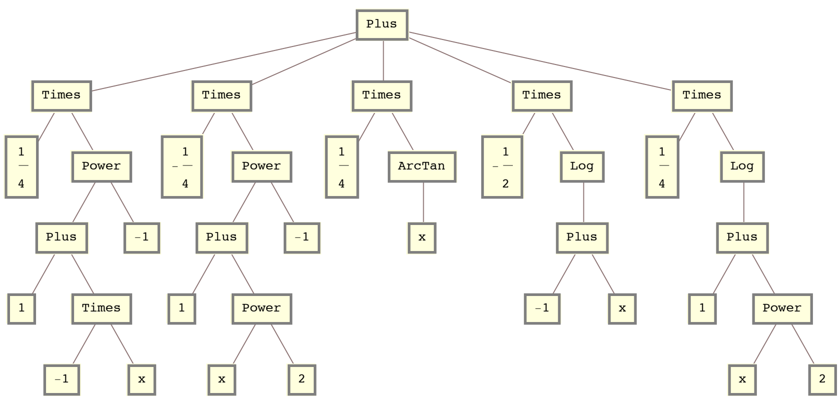
f = ((1 - x)^2 (x^2 + 1)^2)^-1

$$\frac{1}{(1-x)^2\left(1+x^2\right)^2}$$

g = Integrate[f, x]//Expand

$$\frac{1}{4\,(1-x)} - \frac{1}{4\left(1+x^2\right)} + \frac{\text{ArcTan}[x]}{4} - \frac{1}{2}\,\text{Log}[-1+x] + \frac{1}{4}\,\text{Log}\left[1+x^2\right]$$

TreeForm[g]



g3 = Take[g,3]

$$\frac{1}{4\,(1-x)} - \frac{1}{4\left(1+x^2\right)} + \frac{\text{ArcTan}[x]}{4}$$

Part[g, 3]

$$\frac{\text{ArcTan}[x]}{4}$$

```
g12 = Take[g, 2]
```

$$\frac{1}{4\ (1-x)}-\frac{1}{4\ \left(1+x^2\right)}$$

```
Part[g, {4,5} ]
```

$$-\frac{1}{2}\,\text{Log}[-1+x]+\frac{1}{4}\,\text{Log}\left[1+x^2\right]$$