

---

## Demo file to Test EIC YP Structure Function Tables:

Demonstration to show how to use LHAPDF structure function tables with ManeParse.

This is a simpler demo file that just shows the basics of loading the tables.

### Please cite:

ManeParse : A Mathematica reader for Parton Distribution Functions

D.B.Clark, E.Godat, F.I.Olness

Published in : Comput.Phys.Commun .216 (2017) 126 - 137

e - Print : 1605.08012[hep - ph]

Thanks to Tim Hobbs for providing the sample tables.

### BUG: (April 2021: Version 4.0)

In the xxx.info file, input fields (such as "SetDesc:") CANNOT be split across multiple lines.

While this is valid YAML, ManeParse reads a single line (record) at a time.

The "patch" for this is simply remove \newlines.

E.g.,

SetDesc: "This will BREAK \cr  
the ManeParse Program"

SetDesc: "This will NOT break the ManeParse Program"

Fred : 19 April 2021

```
In[1]:= Clear["Global`*"];  
  
In[2]:= $Version  
Out[2]:= 12.1.0 for Linux x86 (64-bit) (March 14, 2020)
```

---

## Set Directory

This example notebook is written with relative directories and is intended to be run within the folder extracted from the tarball.

```
In[3]:= (* This just drops the leading path  
info to make the list of files easier to read *)  
dropPath = Take[(FileNameSplit /@ #) // Transpose, -1][[1]] &;  
  
In[4]:= NotebookDirectory [];  
here = NotebookDirectory []  
  
Out[5]:= /home/olness/Dropbox/mp/ManeParse5_DEMO/FOR_WEB/SF_DEMO_v01/
```

```

In[6]:= (* If there is a problem with the Mathematica working directory,
you can enter it manually here *)
SetDirectory[here]

Out[6]= /home/olness/Dropbox/mp/ManeParse5_DEMO/FOR WEB/SF_DEMO_v01

In[7]:= (* This shows what files should be in this main directory *)
FileNames["*", here] // dropPath

Out[7]= {LHAPDF, MP_packages, testStrFun_v01.nb}

```

---

## Setup Other Directories

```

In[8]:= dirPackages = here <> "MP_packages ";
FileNames["*", dirPackages] // dropPath

Out[9]= {pdfCalc.m, pdfErrors.m, pdfParseCTEQ.m,
pdfParseLHA.m, README_V05.TXT, README_V05.TXT~}

In[10]:= dirFilesLHA = "/usr/local/share/LHAPDF";
dirFilesLHA = here <> "LHAPDF";
dirList = FileNames["*", dirFilesLHA];
dirList // dropPath

Out[13]= {nCTEQ15FullNuc_SF_56_26_nlo}

```

# Load the package

Loading the main package provides many useful functions

```
In[14]:= Get[dirPackages <> "/pdfParseLHA .m"]
```

```
Version : pdfCalc 5.0
Version : ManeParse 5.0: April 2021
- Required Package : pdfCalc --Loaded -
```

```
=====
```

```
- pdfParseLHA -
Version : 5.0: April 2021
Authors : E.J. Godat , D.B. Clark & F.I. Olness
```

```
Please cite: *****
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands , enter : ?pdf*
```

```
=====
```

```
In[15]:= Get[dirPackages <> "/pdfParseCTEQ .m"]
```

```
=====
```

```
- pdfParseCTEQ -
Version : 5.0: April 2021
Authors : D.B. Clark , E.J. Godat & F.I. Olness
```

```
Please cite: *****
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands , enter : ?pdf*
```

```
=====
```

```
In[16]:= Get[dirPackages <> "/pdfErrors.m"]
```

```
=====
```

```
- pdfErrors -
```

```
Version : 5.0; April 2021
```

```
Authors : D.B. Clark, E.J. Godat & F.I. Olness
```

```
Please cite: *****
```

```
http://ncteq.hepforge.org/code/pdf.html
```

```
For a list of available commands, enter: ?pdf*
```

```
=====
```

All functions begin with 'pdf'. To obtain a list of available functions, type the command '?pdf\*'.

```
In[17]:= ? pdf*
```

```
▼ pdfCalc`
```

```
pdfAlphaS
```

```
pdfFunctionX
```

```
pdfGetXlist
```

```
pdfNumQpartit'.  
ion
```

```
pdfSetList
```

```
pdfXmin
```

```
pdfFlavor
```

```
pdfGetInfo
```

```
pdfLowFunction
```

```
pdfReset
```

```
pdfSetListDispl'.  
ay
```

```
pdfFunction
```

```
pdfGetQlist
```

```
pdfLuminosity
```

```
pdfSetInterpol'.  
ator
```

```
pdfSetXpower
```

```
▼ pdfErrors`
```

```
pdfError
```

```
pdfHessianCorrelation
```

```
pdfMCCentral
```

```
pdfMCCorrelation
```

```
pdfFamilyFunction
```

```
pdfHessianError
```

```
pdfMCCentralInterval
```

```
pdfMCError
```

```
▼ pdfParseCTEQ`
```

```
pdfFamilyParseCTEQ
```

```
pdfParseCTEQ
```

```
▼ pdfParseLHA`
```

```
pdfFamilyParseLHA
```

```
pdfParseLHA
```

```
Out[17]=
```

## Individual file manipulation

In[18]:= **pdfReset []**

Default Mathematica interpolator will be used.

All internal variables have been reset.

Individual files in either LHA or PDS format can be parsed using the functions loaded from the packages. Here we demonstrate the LHA parsing function

In[19]:= **? pdfParseLHA**

Symbol

**pdfParseLHA** [fileNameInfo , fileNameData , [verbose ]]: This function reads an individual .info file and .data file specified by *fileNameInfo* and *fileNameData* , respectively , into memory .

The function returns a set number that corresponds to the listing of the .dat file in *pdfSetList* .

Out[19]=

Additionally , the function checks that the  
number and the order of the flavors are the same in both files .

The optional input allows the user to suppress  
the output of this function by choosing *verbose* to be *False* .

In[20]:= **fileList = Select[dirList , (StringMatchQ[#, "\*nCTEQ\*"] &)]**

Out[20]= {/home/olness/Dropbox/mp/ManeParse5\_DEMO /FOR  
WEB/SF\_DEMO\_v01//LHAPDF/nCTEQ15FullNuc\_SF \_56\_26\_nlo}

In[21]:= **files = FileNames["\*", fileList[[1]]];**

**files // dropPath**

Out[22]= {nCTEQ15FullNuc\_SF \_56\_26\_nlo\_00000.dat, nCTEQ15FullNuc\_SF \_56\_26\_nlo.info}

```
In[23]:= fileInfo = Select[files, !StringFreeQ[#, ".info"] &] // First
fileDat = Select[files, !StringFreeQ[#, "0000.dat"] &] // First
```

```
Out[23]:= /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo.info
```

```
Out[24]:= /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo_0000.
dat
```

```
In[25]:= pdfParseLHA [fileInfo , fileDat]
```

```
Successfully read /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo.info .
Successfully read /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo_0000.dat .
```

```
Out[25]= 1
```

```
In[26]:= ? pdfFamilyParseLHA
```

Symbol

pdfFamilyParseLHA [path, [fileType]]: This function reads all the files of type *fileType* in the directory *path* and stores them in memory .

The function returns a list of set numbers that can be used to define a list. These set numbers correspond to the listing of the .dat files in *pdfSetList* .

```
Out[26]= The optional input fileType has a default value of "*.dat".
```

Example :

pdfFamilyParseLHA ["MyGrids ", "ct10\*.dat"] reads all .dat files in the subdirectory "MyGrids " beginning with "ct10" into memory .

```
In[27]:= (* Here is an alternate method to load multiple files *)
pdfFamilyParseLHA [dirList[[1]]]
```

```
Successfully read /home/olness/Dropbox/mp/ManeParse5_DEMO /FOR
WEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo.info .
Included 1 files in the PDF family .
```

```
Out[27]= {2}
```

In[28]:= pdfSetListDisplay []

Set Number	File Name	Max Flavors	Valance Flavors
1	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo_0000.dat	5	n/a
2	/home/olness/Dropbox/mp/ManeParse5_DEMO/FORWEB/SF_DEMO_v01/LHAPDF/nCTEQ15FullNuc_SF_56_26_nlo/nCTEQ15FullNuc_SF_56_26_nlo_0000.dat	5	n/a

## Test Structure Functions

The function “pdf” is left to be defined by the user. Access to the PDF of the set is given by pdfFunction. The function has the canonical form: pdfFunction [setNumber , flavorNumber , x, Q]. If the function is not defined, pdfFunction returns NULL.

In[29]:= ? pdfFunction

Symbol
<p>pdfFunction [setNumber , flavor , x, Q]: This function returns the interpolated value of the PDF for the .pds/.dat file specified by setNumber , for the given flavor and value of Bjorken x and scale Q .</p> <p><i>Warning</i> : The results of this function are only reliable between the maximum and minimum values of x and Q in the .pds/.dat file.</p>

In[30]:= (\* Note, if the flavor is undefined, it will return zero \*)

```
{pdfFunction[1, 900, .1, 10],
 pdfFunction[1, 1, .1, 10]}
```

Out[30]:= {3.82613 , 0.}

In[31]:= Clear[pdf]

```
pdf[iset_?IntegerQ, ipart_?IntegerQ, x_?NumericQ, q_?NumericQ] :=
 pdfFunction[iset, ipart, x, q]
SetAttributes[pdf, Listable]
```

```
In[34]:= pdfGetInfo[1] // TableForm
```

```
Out[34]/TableForm=
```

```
SetDesc → ' Structure functions generated with nCTEQ15 PDFs: S-ACOT(chi) '
SetIndex → 0
Authors → Brandon Stevenson, Lucas Koltz, Tim Hobbs, Fred Olness
Reference → arXiv:1509.00792 nCTEQ15 nPDFs
Format → lhagrid1
DataVersion → 1
NumMembers → 1
Particle → 2212
Flavors → {-930, 930, 932, 931, -900, 900, -901, 901, -940, 940, 942, 941, -908, 908, 910,
Names → {"F1_wm", "F2_wm", "F3_wm", "FL_wm", "F1_gam", "F2_gam", "F3_gam", "FL_gam", "F1_wp", '
OrderQCD → 1
FlavorScheme → variable
NumFlavors → 5
ErrorType → no error
XMin →  $5. \times 10^{-6}$ 
XMax → 1.
QMin → 1.3
QMax → 10 000.
MZ → 91.188
MUp → 0.0
MDown → 0.0
MStrange → 0.0
MCharm → 1.3
MBottom → 4.5
MTop → 174.0
AlphaS_MZ → 1.179973 e-01
AlphaS_OrderQCD → 1
AlphaS_Type → ipol
AlphaS_Qs → {1.3, 1.6259, 2.03351, 2.5433, 3.18089, 3.97833, 4.97567, 6.22305, 7.78314, 9
AlphaS_Vals → {0.39676, 0.34313, 0.30389, 0.27365, 0.2495, 0.22957, 0.21357, 0.20084, 0.1
```

```
In[35]:= flavors = "Flavors" /. pdfGetInfo[1]
```

```
Out[35]= {-930, 930, 932, 931, -900, 900, -901, 901, -940, 940, 942, 941, -908, 908, 910, 909,
-933, 933, 935, 934, -911, 911, 913, 912, -943, 943, 945, 944, -914, 914, 916, 915,
-936, 936, 938, 937, -917, 917, 919, 918, -946, 946, 948, 947, -920, 920, 922, 921}
```

```
In[36]:= pdf[1, flavors, 0.1, 10]
```

```
Out[36]= {66.8715, 14.137, 70.2044, 0.763743, 18.128, 3.82613, 0., 0.200815, 68.626, 14.5005,
86.607, 0.77644, 18.1405, 3.82877, 0.326412, 0.200956, 7.75062, 1.74679, -7.3805,
0.196119, 0.84, 0.196896, 0., 0.02877, 9.68779, 2.14869, 11.2134, 0.210692,
0.840359, 0.19698, -2.80047  $\times 10^{-20}$ , 0.0287828, 0.00239303, 0.00062159, 0.00327363,
0.000142603, 0.0139534, 0.00330421, 0., 0.000508361, 0.0020013, 0.000540554,
-0.00249702, 0.00013989, 0.0139755, 0.00331001, 6.7661  $\times 10^{-21}$ , 0.000509736}
```

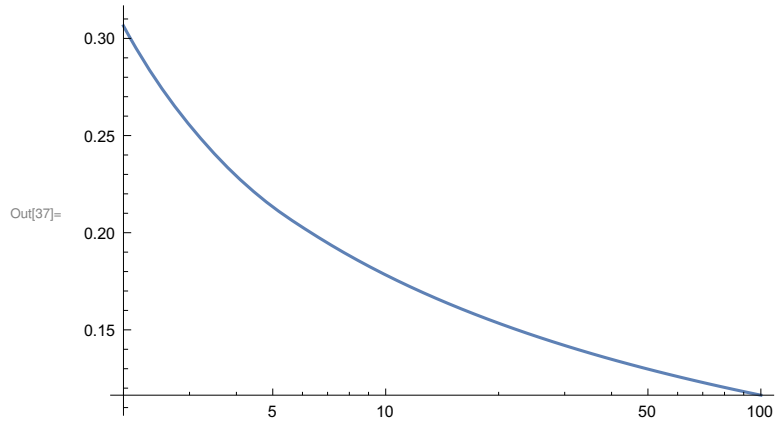


## Example: Alpha-S

In[37]:= **LogLinearPlot** [pdfAlphaS[1, q], {q, 2, 100}]

Created pdfAlphaS for iSet = 1

1 has 1 sub-grid



## Example: Plotting Single Functions

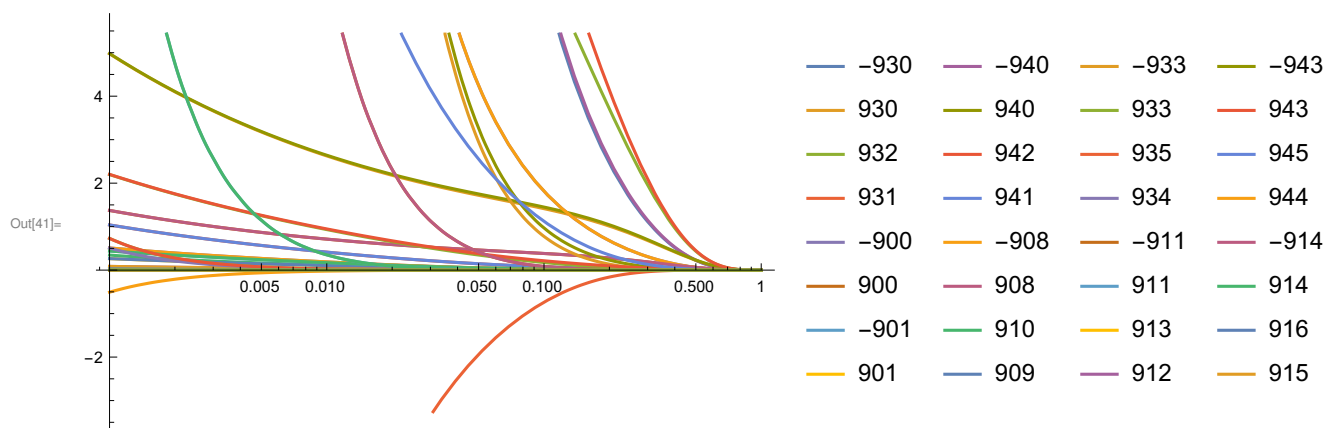
In[38]:= **flavors**

Out[38]= {-930, 930, 932, 931, -900, 900, -901, 901, -940, 940, 942, 941, -908, 908, 910, 909, -933, 933, 935, 934, -911, 911, 913, 912, -943, 943, 945, 944, -914, 914, 916, 915, -936, 936, 938, 937, -917, 917, 919, 918, -946, 946, 948, 947, -920, 920, 922, 921}

In[39]:= **Clear**[x];

**q0 = 10. ;**

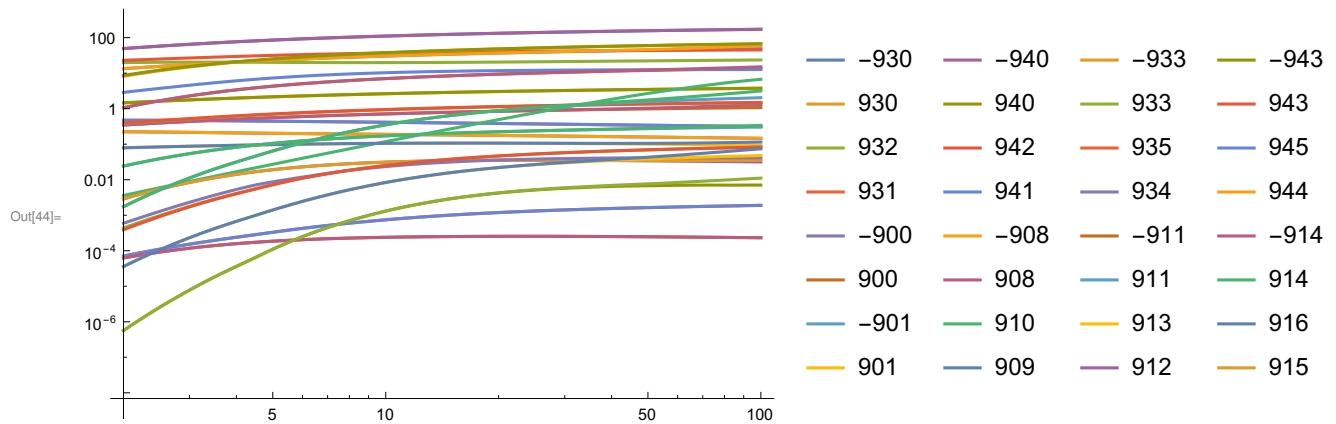
**LogLinearPlot** [x pdf[1, flavors, x, q0] // Evaluate, {x, 10<sup>-3</sup>, 1}, PlotLegends → flavors]



```

In[42]:= Clear[q];
x0 = 0.01 ;
LogLogPlot[x0 pdf[1, flavors, x0, q] // Evaluate, {q, 2, 100}, PlotLegends -> flavors]

```



## Example: 3D Plots

```

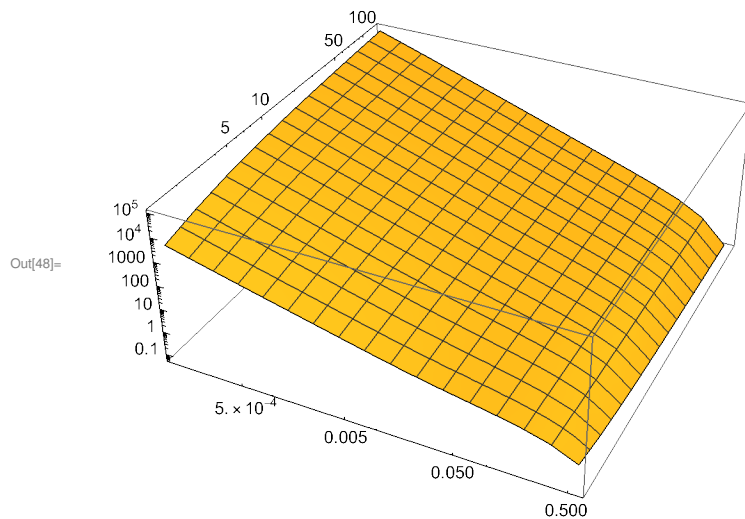
In[45]:= xlist = Table[10. ^ i, {i, -4, 0, 1/4}];
qlist = Table[1.3 * 10. ^ i, {i, 0, 2, 1/4}];
data = Table[{x = xlist[[i]], q = qlist[[j]], x pdf[1, flavors[[1]], x, q]},
  {i, 1, Length[xlist]}, {j, 1, Length[qlist]}];

```

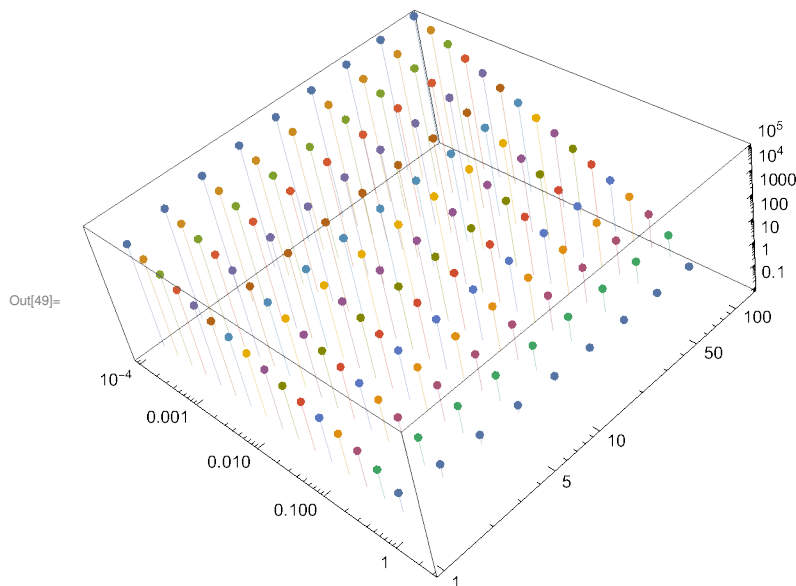
```

In[48]:= ListPlot3D[data // Flatten[#, 1] &, ScalingFunctions -> {"Log", "Log", "Log"}]

```



```
In[49]:= ListPointPlot3D[data, ScalingFunctions → {"Log", "Log", "Log"},
      Filling → Bottom]
```



**Note: the below ONLY works if the “Names” are defined in the info file.**

## getNames Function

```
In[50]:= getNames[iset_] := Module[{names, stream, word, word2, expression},
      names = "Names" /. pdfGetInfo[iset];
      stream = names // StringToStream;
      word = Read[stream, Word];
      word2 = StringReplace[word, {"[" → "{", "]" → "}"}];
      expression = ToExpression[word2];
      Return[expression]
    ]
```

```
In[51]:= getNames[1]
```

```
Out[51]= {F1_wm, F2_wm, F3_wm, FL_wm, F1_gam, F2_gam, F3_gam, FL_gam, F1_wp, F2_wp, F3_wp,
      FL_wp, F1_g+Z, F2_g+Z, F3_g+Z, FL_g+Z, F1c_wm, F2c_wm, F3c_wm, FLc_wm, F1c_gam,
      F2c_gam, F3c_gam, FLc_gam, F1c_wp, F2c_wp, F3c_wp, FLc_wp, F1c_g+Z, F2c_g+Z,
      F3c_g+Z, FLc_g+Z, F1b_wm, F2b_wm, F3b_wm, FLb_wm, F1b_gam, F2b_gam, F3b_gam,
      FLb_gam, F1b_wp, F2b_wp, F3b_wp, FLb_wp, F1b_g+Z, F2b_g+Z, F3b_g+Z, FLb_g+Z}
```

## getFNames Function

```
In[52]:= Clear[getFname];
fnamesTmp = getNames[1]
flavors = "Flavors" /. pdfGetInfo[1]
Do[
  n = flavors[[i]];
  tmp = fnamesTmp[[i]];
  getFname[n] = tmp;
  (* Print[ n, " ", fname[n]]; *)
, {i, 1, Length[flavors]}]
```

```
Out[53]= {F1_wm, F2_wm, F3_wm, FL_wm, F1_gam, F2_gam, F3_gam, FL_gam, F1_wp, F2_wp, F3_wp,
  FL_wp, F1_g+Z, F2_g+Z, F3_g+Z, FL_g+Z, F1c_wm, F2c_wm, F3c_wm, FLc_wm, F1c_gam,
  F2c_gam, F3c_gam, FLc_gam, F1c_wp, F2c_wp, F3c_wp, FLc_wp, F1c_g+Z, F2c_g+Z,
  F3c_g+Z, FLc_g+Z, F1b_wm, F2b_wm, F3b_wm, FLb_wm, F1b_gam, F2b_gam, F3b_gam,
  FLb_gam, F1b_wp, F2b_wp, F3b_wp, FLb_wp, F1b_g+Z, F2b_g+Z, F3b_g+Z, FLb_g+Z}
```

```
Out[54]= {-930, 930, 932, 931, -900, 900, -901, 901, -940, 940, 942, 941, -908, 908, 910, 909,
  -933, 933, 935, 934, -911, 911, 913, 912, -943, 943, 945, 944, -914, 914, 916, 915,
  -936, 936, 938, 937, -917, 917, 919, 918, -946, 946, 948, 947, -920, 920, 922, 921}
```

```
In[56]:= SetAttributes[getFname, Listable]
getFname[flavors]
```

```
Out[57]= {F1_wm, F2_wm, F3_wm, FL_wm, F1_gam, F2_gam, F3_gam, FL_gam, F1_wp, F2_wp, F3_wp,
  FL_wp, F1_g+Z, F2_g+Z, F3_g+Z, FL_g+Z, F1c_wm, F2c_wm, F3c_wm, FLc_wm, F1c_gam,
  F2c_gam, F3c_gam, FLc_gam, F1c_wp, F2c_wp, F3c_wp, FLc_wp, F1c_g+Z, F2c_g+Z,
  F3c_g+Z, FLc_g+Z, F1b_wm, F2b_wm, F3b_wm, FLb_wm, F1b_gam, F2b_gam, F3b_gam,
  FLb_gam, F1b_wp, F2b_wp, F3b_wp, FLb_wp, F1b_g+Z, F2b_g+Z, F3b_g+Z, FLb_g+Z}
```

```
In[58]:= getFname[900]
```

```
Out[58]= F2_gam
```

```
In[59]:= {flavors, getFname[flavors]} // Transpose // Grid[#, Frame → All] &
```

-930	F1_wm
930	F2_wm
932	F3_wm
931	FL_wm
-900	F1_gam
900	F2_gam
-901	F3_gam
901	FL_gam
-940	F1_wp
940	F2_wp

Out[59]=

942	F3_wp
941	FL_wp
-908	F1_g+Z
908	F2_g+Z
910	F3_g+Z
909	FL_g+Z
-933	F1c_wm
933	F2c_wm
935	F3c_wm
934	FLc_wm
-911	F1c_gam
911	F2c_gam
913	F3c_gam
912	FLc_gam
-943	F1c_wp
943	F2c_wp
945	F3c_wp
944	FLc_wp
-914	F1c_g+Z
914	F2c_g+Z
916	F3c_g+Z
915	FLc_g+Z
-936	F1b_wm
936	F2b_wm
938	F3b_wm
937	FLb_wm
-917	F1b_gam
917	F2b_gam
919	F3b_gam
918	FLb_gam
-946	F1b_wp
946	F2b_wp
948	F3b_wp
947	FLb_wp
-920	F1b_g+Z
920	F2b_g+Z
922	F3b_g+Z
921	FLb_g+Z