ParseArg

A command-line argument parser

ParseArg is inspired by arg.h from "*Plan 9 from Outer Space*".

Function

```
ParseArg[args_, pats_] :=
  Block[{CurrentFlag, CurrentToken, ReadString, ReadNumber, ReadChar},
   Module[{as, cs, flg, tok},
    CurrentFlag[] := flg;
    CurrentToken[] := tok;
    ReadChar[] :=
     If[(Length[cs] > 1),
      cs = Rest[cs]; First[cs],
      Throw["Expect more characters after " <> First[as], Symbol["EParseArg"]]
     ];
    ReadNumber[] :=
     If[(Length[cs] > 1),
      With[{s = TakeWhile[Rest[cs], StringMatchQ[#, DigitCharacter] &]},
       If[(Length[s] > 0),
        cs = Drop[cs, Length[s]]; FromDigits[StringJoin[s]],
        Throw["Require a number after " <> First[as], Symbol["EParseArg"]]
       ]],
      If[(Length[as] > 1),
       as = Rest[as];
       If[StringMatchQ[First[as], NumberString],
        FromDigits[First[as]],
        Throw["Expected a number, but got " <> First[as], Symbol["EParseArg"]]
       ],
       Throw["Require a number after " <> First[as], Symbol["EParseArg"]]
      ]
     ];
    ReadString[] :=
     If[(Length[cs] > 1),
      With[{str = StringJoin[Rest[cs]]},
       cs = {Last[cs]}; str],
      If[(Length[as] > 1),
       as = Rest[as]; First[as],
       Throw["Expect a token after " <> First[as], Symbol["EParseArg"]]
      ]
     ];
```

```
    For[as = args, Length[as] > 0, as = Rest[as],
     tok = First[as];
     flg = "";
     cs = Characters[First[as]];
     If[(First[cs] ≠ "-"),
       "default" /. pats;,
       For[cs = Rest[cs], Length[cs] > 0, cs = Rest[cs],
        flg = First[cs];
        flg /. pats;
       ]
      ]
     ]
    ]];
```

Example

```
Catch[
 ParseArg[{
    "input1.txt",
    "-a",
    "-b", "bird", "-bcat",
    "-c", "100", "-c200",
    "-c300bdog",
    "input2.txt",
    "-di30",
    "-dsyay",
    "-f", "-c999"},
   {
    "a" :→ Print["Opt 'a'"],
    "b" :→ Print["Opt 'b' with string '" <> ReadString[] <> "'"],
    "c" :→ Print["Opt 'c' with number '" <> ToString[ReadNumber[]] <> "'"],
    "d" :→ With[{c = ReadChar[]},
      Which[
        (c == "i"), Print["Opt 'd' with number '" <> ToString[ReadNumber[]] <> "'"],
        (c == "s"), Print["Opt 'd' with string '" <> ReadString[] <> "'"],
        (True), Throw["oops!"]
       ]],
    "default" :→ Print["Argument '" <> CurrentToken[] <> "'"],
    _ :→ Throw["Invalid flag '" <> CurrentFlag[] <> "'", EParseArg]
   }
  ]
 , EParseArg, Print["Parse Error: ", #1] &]
```

```
Argument 'input1.txt'

Opt 'a'

Opt 'b' with string 'bird'

Opt 'b' with string 'cat'

Opt 'c' with number '100'

Opt 'c' with number '200'

Opt 'c' with number '300'

Opt 'b' with string 'dog'

Argument 'input2.txt'

Opt 'd' with number '30'

Opt 'd' with string 'yay'

Parse Error: Invalid flag 'f'
```

License