

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", "-bcats",
    "-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

Copyright © 2014 Eon S. Jeon <esjeon@hyunmu.am>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.