|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $IFS  internal field separator  This variable determines how Bash recognizes [fields](https://www.tldp.org/LDP/abs/html/special-chars.html#FIELDREF), or word boundaries, when it interprets character strings.  $IFS defaults to [whitespace](https://www.tldp.org/LDP/abs/html/special-chars.html#WHITESPACEREF) (space, tab, and newline), but may be changed, for example, to parse a comma-separated data file. Note that [$\*](https://www.tldp.org/LDP/abs/html/internalvariables.html#APPREF) uses the first character held in $IFS. See [Example 5-1](https://www.tldp.org/LDP/abs/html/quotingvar.html#WEIRDVARS).   |  | | --- | | bash$ **echo "$IFS"**  (With $IFS set to default, a blank line displays.)    bash$ **echo "$IFS" | cat -vte**  ^I$  $  (Show whitespace: here a single space, ^I [horizontal tab],  and newline, and display "$" at end-of-line.)  bash$ **bash -c 'set w x y z; IFS=":-;"; echo "$\*"'**  w:x:y:z  (Read commands from string and assign any arguments to pos params.) |   Set $IFS to eliminate whitespace in [pathnames](https://www.tldp.org/LDP/abs/html/special-chars.html#PATHNAMEREF).   |  | | --- | | IFS="$(printf '\n\t')" # Per David Wheeler. |  |  |  |  | | --- | --- | --- | | Caution | $IFS does not handle whitespace the same as it does other characters.  **Example 9-1. $IFS and whitespace**   |  | | --- | | #!/bin/bash  # ifs.sh  var1="a+b+c"  var2="d-e-f"  var3="g,h,i"  IFS=+  # The plus sign will be interpreted as a separator.  echo $var1 # a b c  echo $var2 # d-e-f  echo $var3 # g,h,i  echo  IFS="-"  # The plus sign reverts to default interpretation.  # The minus sign will be interpreted as a separator.  echo $var1 # a+b+c  echo $var2 # d e f  echo $var3 # g,h,i  echo  IFS=","  # The comma will be interpreted as a separator.  # The minus sign reverts to default interpretation.  echo $var1 # a+b+c  echo $var2 # d-e-f  echo $var3 # g h i  echo  IFS=" "  # The space character will be interpreted as a separator.  # The comma reverts to default interpretation.  echo $var1 # a+b+c  echo $var2 # d-e-f  echo $var3 # g,h,i  # ======================================================== #  # However ...  # $IFS treats whitespace differently than other characters.  output\_args\_one\_per\_line()  {  for arg  do  echo "[$arg]"  done # ^ ^ Embed within brackets, for your viewing pleasure.  }  echo; echo "IFS=\" \""  echo "-------"  IFS=" "  var=" a b c "  # ^ ^^ ^^^  output\_args\_one\_per\_line $var # output\_args\_one\_per\_line `echo " a b c "`  # [a]  # [b]  # [c]  echo; echo "IFS=:"  echo "-----"  IFS=:  var=":a::b:c:::" # Same pattern as above,  # ^ ^^ ^^^ #+ but substituting ":" for " " ...  output\_args\_one\_per\_line $var  # []  # [a]  # []  # [b]  # [c]  # []  # []  # Note "empty" brackets.  # The same thing happens with the "FS" field separator in awk.  echo  exit | |   (Many thanks, Stéphane Chazelas, for clarification and above examples.) |