On Jun 22, 8:06 am, JDS <jdtsmith.nos...@yahoo.com> wrote:
> How can you sort a HASH() in IDL?
>
> IDL> h=hash(['a','c','d','b'],[7,1,5,8])
> IDL> print,h
> c: 1
> a: 7
> b: 8
> d: 5
>
> OK, they are randomly sorted. That's cool, it's a hash. I'll just sort on the hash key...
>
> IDL> s=sort(h->keys())
> IDL> print,s
>   0
>
> Hmmm, it seems this isn't sorting correctly...
>
> IDL> help, h->keys()
> <Expression>   LIST <ID=3175 NELEMENTS=4>
>
> Ahah, the return of Keys() is a list! (Why does SORT pretend it can sort a LIST?). We need
an array.   Continuing onwards...
>
> IDL> s=sort((h->keys())->toarray())
> IDL> print,(h.keys())[s]
> a
> b
> c
> d
>
> Now we're getting there. Let's just index...
>
> IDL> print,h[(h.keys())[s]]
> c: 1
> a: 7
> b: 8
> d: 5
>
> Uhhh... what? Maybe it has something to do with indexing via a LIST. Let's make it an
array first:
>
> IDL> print,((h.keys()).toArray())[s]
> a b c d

>
> Should be perfect...
>
> IDL> print,h[((h.keys()).toArray())[s]]
> c: 1
> a: 7
> b: 8
> d: 5
>
> But it's not!
>
> Is there any way to index a hash and have the order of the keys be maintained, for example for this sorting problem? Loss of order is a well known feature of hash storage, but when you explicitly specify the order, it should be respected. Am I missing something?

I don't see my first post of this... so apologies if it appears
twice. I had the same question a little while ago:

 http://groups.google.com/group/comp.lang.idl-pvwave/browse_t
hread/thread/fa55eae4c4be5880/012750823bc9dce2