Improving Typed Data

* It would be nice to have a version of view that takes no arguments and returns a new TypedData with the same offset and length.
* Why are the constructors in TypedData called <Type>.fromList rather than .from
* Because TypedData and ByteData are not List<E> and List<int> respectively you can't write

ByteData toByteData(TypedData vList) {  
 **assert**(vList != **null**);  
 **return** (vList.**isEmpty**) **?**kEmptyByteData;  
 **:** vList.buffer.asByteData(vList.offsetInBytes,vList.lengthInBytes);  
}

* It is painful to constantly include a Endian arg for the ByteData setters and getters. It would be more convenient if the ByteData constructor took an Endian argument.
* It would be convenient if ByteData had methods such as:

getXintNNList(int byteffset, int listLength)

setXintNNList(int offset, TypedDataList list)

# Current Type Hierarchy

TypedData

ByteData

\_TypedIntList extends TypedData

Int8List implements List<int>, \_TypedIntList

…

\_TypedFloatList extends TypedData

Float32List implements List<double>, \_TypedFloatList

# Proposed Modifications

TypedList<V> implements List<V>

TypedIntList implements List<int>

ByteData

Int8List

…

TypedFloatList implements List<double>

Float32List

…

A better named than TypedList might be FixedList<V> or FixedSizeList<V>. This would lead to the following:

FixedList<E> implements List<E>

FixedIntList implements List<int>

ByteData

Int8List

…

FixedFloatList implements List<double>

Float32List

…

Element Type Hierarchy

ElementMixin

Element

Number

Integer

OB, UN

OBbd, UNbd

OBtag, UNtag

SS, US, OW

AT, SL, UL, OL

PN

Float

FL, OF

FD, OD

String

Structured

DS, IS,

AS, DA, DT, TM

UI, UR

Unstructured

CS

SH, LO, UC

ST, LT, UT

Sequence

SQ