Picking apart a floating point number

In [1]:

```
#keep

# Never mind the details of this function...

def pretty_print_fp(x):
    print("---------------------------------------------")
    print("Floating point structure for %r" % x)
    print("---------------------------------------------")
    import struct
    s = struct.pack("d", x)

    def get_bit(i):
        byte_nr, bit_nr = divmod(i, 8)
        return int(bool(
            s[byte_nr] & (1 << bit_nr)
            ))

    def get_bits(lsb, count):
        return sum(get_bit(i+lsb)*2**i for i in range(count))

    # https://en.wikipedia.org/wiki/Double_precision_floating-point_format

    print("                                 1         2         3         4
5")
    print("indices              : 0123456789012345678901234567890123456789012345
67890123456789")
    print("Sign bit (1:negative):", get_bit(63))
    exponent = get_bits(52, 11)
    print("Exponent     (binary):", bin(exponent)[2:])
    print("Exponent    (shifted): %d" % (exponent - 1023))
    fraction = get_bits(0, 52)
    significand = fraction + 2**52
    print("Significand  (binary):", bin(significand)[2:])
    print("Significand (shifted):", repr(significand / (2**52)))
```

In [8]:

```
#keep
pretty_print_fp(2**1024)
```

```
---------------------------------------------
Floating point structure for 17976931348623159077293051907890247336l
79769789423065727343008115773267580550096313270847732240753602112011
38798713933576587897688144166224928474306394741243777678934248654852
76302219601246094119453082952085005768838150682342462881473913110540
8272371633505106845862982399472459384797163048353563296242241372l6
---------------------------------------------

---------------------------------------------------------------------
-------
error                                     Traceback (most recent cal
l last)
<ipython-input-8-46758d327020> in <module>()
      1 #keep
----> 2 pretty_print_fp(2**1024)

<ipython-input-1-0c0d10989b38> in pretty_print_fp(x)
      8     print("---------------------------------------------")
      9     import struct
---> 10     s = struct.pack("d", x)
     11 
     12     def get_bit(i):

error: required argument is not a float
```

Things to try:

- Twiddle the sign bit
- 1,2,4,8
- 0.5,0.25
- 2^{1023}, 2^{1024}
- 2^{-1023}, 2^{-1024}
- `float("nan")`

In [15]:

```
import numpy as np
```

In [16]:

```
np.binary_repr(1024)
```

Out[16]:

```
'10000000000'
```

In [18]:

```
bin(1024+1023)
```

Out[18]:

```
'0b11111111111'
```

In [ ]:

```
```