tf.where

```
where(
    condition,
    x=None,
    y=None,
    name=None
)
```

Defined in `tensorflow/python/ops/array_ops.py`.

See the guides: Control Flow > Comparison Operators, Math > Sequence Comparison and Indexing

Return the elements, either from `x` or `y`, depending on the `condition`.

If both `x` and `y` are None, then this operation returns the coordinates of true elements of `condition`. The coordinates are returned in a 2-D tensor where the first dimension (rows) represents the number of true elements, and the second dimension (columns) represents the coordinates of the true elements. Keep in mind, the shape of the output tensor can vary depending on how many true values there are in input. Indices are output in row-major order.

If both non-None, `x` and `y` must have the same shape. The `condition` tensor must be a scalar if `x` and `y` are scalar. If `x` and `y` are vectors of higher rank, then `condition` must be either a vector with size matching the first dimension of `x`, or must have the same shape as `x`.

The `condition` tensor acts as a mask that chooses, based on the value at each element, whether the corresponding element / row in the output should be taken from `x` (if true) or `y` (if false).

If `condition` is a vector and `x` and `y` are higher rank matrices, then it chooses which row (outer dimension) to copy from `x` and `y`. If `condition` has the same shape as `x` and `y`, then it chooses which element to copy from `x` and `y`.

Args:

- `condition`: A `Tensor` of type `bool`
- `x`: A Tensor which may have the same shape as `condition`. If `condition` is rank 1, `x` may have higher rank, but its first dimension must match the size of `condition`.
- `y`: A `tensor` with the same shape and type as `x`.
- `name`: A name of the operation (optional)

Returns:

A `Tensor` with the same type and shape as `x`, `y` if they are non-None. A `Tensor` with shape `(num_true, dim_size(condition))`.

Raises:

- `ValueError`: When exactly one of `x` or `y` is non-None.

Last updated November 2, 2017.