

Gabriele Petrillo

# Programmazione in Faust

*L'obiettivo di questa lezione è dare una panoramica ampia del mondo di Faust. Non andremo ancora nei dettagli ma vedremo solo cosa è on grado di fare Faust.*

## Faust

Faust è un linguaggio di programmazione funzionale per la sintesi del suono e l'elaborazione dell'audio, particolarmente adatto alla progettazione di sintetizzatori, strumenti musicali ed effetti audio. Faust è utilizzato dal vivo per concerti e produzioni artistiche, nella didattica e nella ricerca, in progetti open source e in applicazioni commerciali. Faust è in grado di scrivere plugin VST e Audio Unit, realizzare applicazioni Android e IOS, externals per vari linguaggi di computer music come Max, PD, SuperCollider, CSOUND, Chuck e altri ancora. A questo proposito, Faust può essere visto come un'alternativa al C++, ma è molto più semplice e intuitivo da imparare.

## Editor Online

Faust può essere programmato attraverso uno strumento online che non richiede alcuna installazione. Il codice di Faust può essere scritto ed eseguito direttamente in un browser web.

L'editor online di Faust utilizza una tecnologia web all'avanguardia come WebAssembly, e le Web audio API. Per questo motivo, è necessario utilizzare un browser web molto recente che supporti WebAssembly<sup>1</sup>. Alcune cose avranno bisogno del supporto MIDI e attualmente solo Google Chrome supporta la web midi API, pertanto, si consiglia di utilizzare questo browser.

Apriamo l'editor di Faust<sup>2</sup> e scriviamo le prime due righe di codice:

```
import("stdfaust.lib");  
process = + ;
```

Queste prime due righe di codice sono molto importanti. La prima importa la libreria standard di Faust, mentre la seconda è la dichiarazione della keyword *process* che rappresenta la funzione main del programma<sup>3</sup>.

Quindi, sostituiamo qui la definizione della linea di processo:

```
import("stdfaust.lib");  
process = button("gate") :  
    pm.djembe(60, 0.3, 0.4, 1);
```

Button è un interfaccia utente primitiva che genera un segnale pari a 1 ogni volta che viene premuta con il mouse. La grammatica è *button("nome dell'oggetto")*. Adesso tramite i *due punti* ( : ) possiamo collegare l'uscita del button ad un modello fisico di un djembe *pm.djembe(60,0.3,0.4,1)*, premendo *run* eseguiremo lo script.

In faust le uscite e gli ingressi fisici sono connessi automaticamente, quindi tutti i circuiti sono connessi ai primi ingressi e/o uscite liberi disponibili.

## Primo programma in Faust

<sup>1</sup>WebAssembly (abbreviato Wasm) è un formato di istruzioni binarie per una virtual-machine stack-based. Wasm è progettato come target portatile per la compilazione di linguaggi di alto livello come *c* e *c++*, consentendo la distribuzione sul web per applicazioni client e server

<sup>2</sup><https://faustide.grame.fr>

<sup>3</sup>In Faust la funzione *process* equivale alla funzione *main* dei linguaggi di programmazione *c* o *c++*

Un programma Faust è una serie di istruzioni che terminano con il *punto e virgola* ( ; ) è importante non scordare mai il punto e virgola, è l'errore più comune quando si inizia in Faust. L'esempio precedente è composto da due istruzioni: la prima importa l'intera libreria standard di Faust. Una libreria è una collezione di circuiti predefiniti scritti in Faust. Queste collezioni sono organizzate in gerarchie di ambienti che possono essere raggiunti tramite un semplice prefisso. Per esempio, nel codice precedente, *pm.* corrisponde a *physical modelling library*.

La seconda è la nostra *process line* che implementa un semplice *djembe*, un modello fisico di una percussione azionata da un bottone. *Button* è un interfaccia utente primitiva<sup>4</sup> che appare come un elemento nell'interfaccia utente grafica. Questo circuito produce un segnale che è zero quando non è premuto, e 1 quando è premuto.

La documentazione di ogni elemento può essere ottenuta semplicemente posizionando il cursore del mouse sopra il nome dell'elemento. Se guardiamo la documentazione di *pm.djembe* possiamo vedere che:

1. Freq;
2. Strike position;
3. Strike sharpness;
4. Gain;
5. Trigger;

Queste rappresentano in ordine gli ingressi del circuito di cui abbiamo definito come parametri i primi quattro e connesso (in maniera automatica visto che era l'unico input vuoto) l'ultimo al *button* tramite il segno dei *due punti* ( : ) che collega due circuiti in sequenza. Esiste anche un'altra tipologia di notazione che forse rende più chiaro quello che abbiamo programmato:

```
import("stdfaust.lib");
process = 60, 0.3, 0.4, 1, button("gate") :
    pm.djembe;
```

<sup>4</sup>Gli elementi primitivi sono elementi predefiniti di un linguaggio

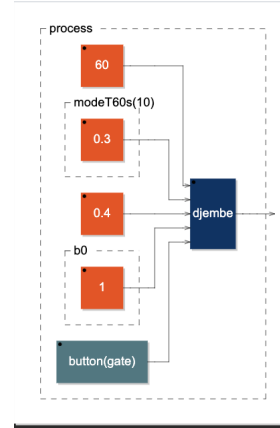


fig.1

### Aggiungiamo un riverbero

Per aggiungere un riverbero al nostro strumento possiamo utilizzare il riverbero digitale *freeverb* della libreria standard:

```
import("stdfaust.lib");
process = 60, 0.3, 0.4, 1, button("gate") :
    pm.djembe : dm.freeverb_demo;
```

Tuttavia questo codice produrrà un errore perché *freeverb* ha due input, mentre *djembe* un solo output. Infatti, compilando il codice, verrà visualizzato un errore di questo tipo:

```
Error

ERROR in sequential composition djembe.freeverb_demo The number of outputs [1] of djembe must be equal to the number of inputs [2] of freeverb_demo Here djembe = \x1.\x2.\x3.\x4.\x5.\x6.\x7.\x8.\x9.\x10.\x11.\x12.\x13.\x14.\x15.\x16.\x17.\x18.\x19.\x20.\x21.\x22.\x23.\x24.\x25.\x26.\x27.\x28.\x29.\x30.\x31.\x32.\x33.\x34.\x35.\x36.\x37.\x38.\x39.\x40.\x41.\x42.\x43.\x44.\x45.\x46.\x47.\x48.\x49.\x50.\x51.\x52.\x53.\x54.\x55.\x56.\x57.\x58.\x59.\x60.\x61.\x62.\x63.\x64.\x65.\x66.\x67.\x68.\x69.\x70.\x71.\x72.\x73.\x74.\x75.\x76.\x77.\x78.\x79.\x80.\x81.\x82.\x83.\x84.\x85.\x86.\x87.\x88.\x89.\x90.\x91.\x92.\x93.\x94.\x95.\x96.\x97.\x98.\x99.\x100.\x101.\x102.\x103.\x104.\x105.\x106.\x107.\x108.\x109.\x110.\x111.\x112.\x113.\x114.\x115.\x116.\x117.\x118.\x119.\x120.\x121.\x122.\x123.\x124.\x125.\x126.\x127.\x128.\x129.\x130.\x131.\x132.\x133.\x134.\x135.\x136.\x137.\x138.\x139.\x140.\x141.\x142.\x143.\x144.\x145.\x146.\x147.\x148.\x149.\x150.\x151.\x152.\x153.\x154.\x155.\x156.\x157.\x158.\x159.\x160.\x161.\x162.\x163.\x164.\x165.\x166.\x167.\x168.\x169.\x170.\x171.\x172.\x173.\x174.\x175.\x176.\x177.\x178.\x179.\x180.\x181.\x182.\x183.\x184.\x185.\x186.\x187.\x188.\x189.\x190.\x191.\x192.\x193.\x194.\x195.\x196.\x197.\x198.\x199.\x200.\x201.\x202.\x203.\x204.\x205.\x206.\x207.\x208.\x209.\x210.\x211.\x212.\x213.\x214.\x215.\x216.\x217.\x218.\x219.\x220.\x221.\x222.\x223.\x224.\x225.\x226.\x227.\x228.\x229.\x230.\x231.\x232.\x233.\x234.\x235.\x236.\x237.\x238.\x239.\x240.\x241.\x242.\x243.\x244.\x245.\x246.\x247.\x248.\x249.\x250.\x251.\x252.\x253.\x254.\x255.\x256.\x257.\x258.\x259.\x260.\x261.\x262.\x263.\x264.\x265.\x266.\x267.\x268.\x269.\x270.\x271.\x272.\x273.\x274.\x275.\x276.\x277.\x278.\x279.\x280.\x281.\x282.\x283.\x284.\x285.\x286.\x287.\x288.\x289.\x290.\x291.\x292.\x293.\x294.\x295.\x296.\x297.\x298.\x299.\x300.\x301.\x302.\x303.\x304.\x305.\x306.\x307.\x308.\x309.\x310.\x311.\x312.\x313.\x314.\x315.\x316.\x317.\x318.\x319.\x320.\x321.\x322.\x323.\x324.\x325.\x326.\x327.\x328.\x329.\x330.\x331.\x332.\x333.\x334.\x335.\x336.\x337.\x338.\x339.\x340.\x341.\x342.\x343.\x344.\x345.\x346.\x347.\x348.\x349.\x350.\x351.\x352.\x353.\x354.\x355.\x356.\x357.\x358.\x359.\x360.\x361.\x362.\x363.\x364.\x365.\x366.\x367.\x368.\x369.\x370.\x371.\x372.\x373.\x374.\x375.\x376.\x377.\x378.\x379.\x380.\x381.\x382.\x383.\x384.\x385.\x386.\x387.\x388.\x389.\x390.\x391.\x392.\x393.\x394.\x395.\x396.\x397.\x398.\x399.\x400.\x401.\x402.\x403.\x404.\x405.\x406.\x407.\x408.\x409.\x410.\x411.\x412.\x413.\x414.\x415.\x416.\x417.\x418.\x419.\x420.\x421.\x422.\x423.\x424.\x425.\x426.\x427.\x428.\x429.\x430.\x431.\x432.\x433.\x434.\x435.\x436.\x437.\x438.\x439.\x440.\x441.\x442.\x443.\x444.\x445.\x446.\x447.\x448.\x449.\x450.\x451.\x452.\x453.\x454.\x455.\x456.\x457.\x458.\x459.\x460.\x461.\x462.\x463.\x464.\x465.\x466.\x467.\x468.\x469.\x470.\x471.\x472.\x473.\x474.\x475.\x476.\x477.\x478.\x479.\x480.\x481.\x482.\x483.\x484.\x485.\x486.\x487.\x488.\x489.\x490.\x491.\x492.\x493.\x494.\x495.\x496.\x497.\x498.\x499.\x500.\x501.\x502.\x503.\x504.\x505.\x506.\x507.\x508.\x509.\x510.\x511.\x512.\x513.\x514.\x515.\x516.\x517.\x518.\x519.\x520.\x521.\x522.\x523.\x524.\x525.\x526.\x527.\x528.\x529.\x530.\x531.\x532.\x533.\x534.\x535.\x536.\x537.\x538.\x539.\x540.\x541.\x542.\x543.\x544.\x545.\x546.\x547.\x548.\x549.\x550.\x551.\x552.\x553.\x554.\x555.\x556.\x557.\x558.\x559.\x560.\x561.\x562.\x563.\x564.\x565.\x566.\x567.\x568.\x569.\x570.\x571.\x572.\x573.\x574.\x575.\x576.\x577.\x578.\x579.\x580.\x581.\x582.\x583.\x584.\x585.\x586.\x587.\x588.\x589.\x590.\x591.\x592.\x593.\x594.\x595.\x596.\x597.\x598.\x599.\x600.\x601.\x602.\x603.\x604.\x605.\x606.\x607.\x608.\x609.\x610.\x611.\x612.\x613.\x614.\x615.\x616.\x617.\x618.\x619.\x620.\x621.\x622.\x623.\x624.\x625.\x626.\x627.\x628.\x629.\x630.\x631.\x632.\x633.\x634.\x635.\x636.\x637.\x638.\x639.\x640.\x641.\x642.\x643.\x644.\x645.\x646.\x647.\x648.\x649.\x650.\x651.\x652.\x653.\x654.\x655.\x656.\x657.\x658.\x659.\x660.\x661.\x662.\x663.\x664.\x665.\x666.\x667.\x668.\x669.\x670.\x671.\x672.\x673.\x674.\x675.\x676.\x677.\x678.\x679.\x680.\x681.\x682.\x683.\x684.\x685.\x686.\x687.\x688.\x689.\x690.\x691.\x692.\x693.\x694.\x695.\x696.\x697.\x698.\x699.\x700.\x701.\x702.\x703.\x704.\x705.\x706.\x707.\x708.\x709.\x710.\x711.\x712.\x713.\x714.\x715.\x716.\x717.\x718.\x719.\x720.\x721.\x722.\x723.\x724.\x725.\x726.\x727.\x728.\x729.\x730.\x731.\x732.\x733.\x734.\x735.\x736.\x737.\x738.\x739.\x740.\x741.\x742.\x743.\x744.\x745.\x746.\x747.\x748.\x749.\x750.\x751.\x752.\x753.\x754.\x755.\x756.\x757.\x758.\x759.\x760.\x761.\x762.\x763.\x764.\x765.\x766.\x767.\x768.\x769.\x770.\x771.\x772.\x773.\x774.\x775.\x776.\x777.\x778.\x779.\x780.\x781.\x782.\x783.\x784.\x785.\x786.\x787.\x788.\x789.\x790.\x791.\x792.\x793.\x794.\x795.\x796.\x797.\x798.\x799.\x800.\x801.\x802.\x803.\x804.\x805.\x806.\x807.\x808.\x809.\x810.\x811.\x812.\x813.\x814.\x815.\x816.\x817.\x818.\x819.\x820.\x821.\x822.\x823.\x824.\x825.\x826.\x827.\x828.\x829.\x830.\x831.\x832.\x833.\x834.\x835.\x836.\x837.\x838.\x839.\x840.\x841.\x842.\x843.\x844.\x845.\x846.\x847.\x848.\x849.\x850.\x851.\x852.\x853.\x854.\x855.\x856.\x857.\x858.\x859.\x860.\x861.\x862.\x863.\x864.\x865.\x866.\x867.\x868.\x869.\x870.\x871.\x872.\x873.\x874.\x875.\x876.\x877.\x878.\x879.\x880.\x881.\x882.\x883.\x884.\x885.\x886.\x887.\x888.\x889.\x890.\x891.\x892.\x893.\x894.\x895.\x896.\x897.\x898.\x899.\x900.\x901.\x902.\x903.\x904.\x905.\x906.\x907.\x908.\x909.\x910.\x911.\x912.\x913.\x914.\x915.\x916.\x917.\x918.\x919.\x920.\x921.\x922.\x923.\x924.\x925.\x926.\x927.\x928.\x929.\x930.\x931.\x932.\x933.\x934.\x935.\x936.\x937.\x938.\x939.\x940.\x941.\x942.\x943.\x944.\x945.\x946.\x947.\x948.\x949.\x950.\x951.\x952.\x953.\x954.\x955.\x956.\x957.\x958.\x959.\x960.\x961.\x962.\x963.\x964.\x965.\x966.\x967.\x968.\x969.\x970.\x971.\x972.\x973.\x974.\x975.\x976.\x977.\x978.\x979.\x980.\x981.\x982.\x983.\x984.\x985.\x986.\x987.\x988.\x989.\x990.\x991.\x992.\x993.\x994.\x995.\x996.\x997.\x998.\x999.\x1000.\x1001.\x1002.\x1003.\x1004.\x1005.\x1006.\x1007.\x1008.\x1009.\x1010.\x1011.\x1012.\x1013.\x1014.\x1015.\x1016.\x1017.\x1018.\x1019.\x1020.\x1021.\x1022.\x1023.\x1024.\x1025.\x1026.\x1027.\x1028.\x1029.\x1030.\x1031.\x1032.\x1033.\x1034.\x1035.\x1036.\x1037.\x1038.\x1039.\x1040.\x1041.\x1042.\x1043.\x1044.\x1045.\x1046.\x1047.\x1048.\x1049.\x1050.\x1051.\x1052.\x1053.\x1054.\x1055.\x1056.\x1057.\x1058.\x1059.\x1060.\x1061.\x1062.\x1063.\x1064.\x1065.\x1066.\x1067.\x1068.\x1069.\x1070.\x1071.\x1072.\x1073.\x1074.\x1075.\x1076.\x1077.\x1078.\x1079.\x1080.\x1081.\x1082.\x1083.\x1084.\x1085.\x1086.\x1087.\x1088.\x1089.\x1090.\x1091.\x1092.\x1093.\x1094.\x1095.\x1096.\x1097.\x1098.\x1099.\x1100.\x1101.\x1102.\x1103.\x1104.\x1105.\x1106.\x1107.\x1108.\x1109.\x1110.\x1111.\x1112.\x1113.\x1114.\x1115.\x1116.\x1117.\x1118.\x1119.\x1120.\x1121.\x1122.\x1123.\x1124.\x1125.\x1126.\x1127.\x1128.\x1129.\x1130.\x1131.\x1132.\x1133.\x1134.\x1135.\x1136.\x1137.\x1138.\x1139.\x1140.\x1141.\x1142.\x1143.\x1144.\x1145.\x1146.\x1147.\x1148.\x1149.\x1150.\x1151.\x1152.\x1153.\x1154.\x1155.\x1156.\x1157.\x1158.\x1159.\x1160.\x1161.\x1162.\x1163.\x1164.\x1165.\x1166.\x1167.\x1168.\x1169.\x1170.\x1171.\x1172.\x1173.\x1174.\x1175.\x1176.\x1177.\x1178.\x1179.\x1180.\x1181.\x1182.\x1183.\x1184.\x1185.\x1186.\x1187.\x1188.\x1189.\x1190.\x1191.\x1192.\x1193.\x1194.\x1195.\x1196.\x1197.\x1198.\x1199.\x1200.\x1201.\x1202.\x1203.\x1204.\x1205.\x1206.\x1207.\x1208.\x1209.\x1210.\x1211.\x1212.\x1213.\x1214.\x1215.\x1216.\x1217.\x1218.\x1219.\x1220.\x1221.\x1222.\x1223.\x1224.\x1225.\x1226.\x1227.\x1228.\x1229.\x1230.\x1231.\x1232.\x1233.\x1234.\x1235.\x1236.\x1237.\x1238.\x1239.\x1240.\x1241.\x1242.\x1243.\x1244.\x1245.\x1246.\x1247.\x1248.\x1249.\x1250.\x1251.\x1252.\x1253.\x1254.\x1255.\x1256.\x1257.\x1258.\x1259.\x1260.\x1261.\x1262.\x1263.\x1264.\x1265.\x1266.\x1267.\x1268.\x1269.\x1270.\x1271.\x1272.\x1273.\x1274.\x1275.\x1276.\x1277.\x1278.\x1279.\x1280.\x1281.\x1282.\x1283.\x1284.\x1285.\x1286.\x1287.\x1288.\x1289.\x1290.\x1291.\x1292.\x1293.\x1294.\x1295.\x1296.\x1297.\x1298.\x1299.\x1300.\x1301.\x1302.\x1303.\x1304.\x1305.\x1306.\x1307.\x1308.\x1309.\x1310.\x1311.\x1312.\x1313.\x1314.\x1315.\x1316.\x1317.\x1318.\x1319.\x1320.\x1321.\x1322.\x1323.\x1324.\x1325.\x1326.\x1327.\x1328.\x1329.\x1330.\x1331.\x1332.\x1333.\x1334.\x1335.\x1336.\x1337.\x1338.\x1339.\x1340.\x1341.\x1342.\x1343.\x1344.\x1345.\x1346.\x1347.\x1348.\x1349.\x1350.\x1351.\x1352.\x1353.\x1354.\x1355.\x1356.\x1357.\x1358.\x1359.\x1360.\x1361.\x1362.\x1363.\x1364.\x1365.\x1366.\x1367.\x1368.\x1369.\x1370.\x1371.\x1372.\x1373.\x1374.\x1375.\x1376.\x1377.\x1378.\x1379.\x1380.\x1381.\x1382.\x1383.\x1384.\x1385.\x1386.\x1387.\x1388.\x1389.\x1390.\x1391.\x1392.\x1393.\x1394.\x1395.\x1396.\x1397.\x1398.\x1399.\x1400.\x1401.\x1402.\x1403.\x1404.\x1405.\x1406.\x1407.\x1408.\x1409.\x1410.\x1411.\x1412.\x1413.\x1414.\x1415.\x1416.\x1417.\x1418.\x1419.\x1420.\x1421.\x1422.\x1423.\x1424.\x1425.\x1426.\x1427.\x1428.\x1429.\x1430.\x1431.\x1432.\x1433.\x1434.\x1435.\x1436.\x1437.\x1438.\x1439.\x1440.\x1441.\x1442.\x1443.\x1444.\x1445.\x1446.\x1447.\x1448.\x1449.\x1450.\x1451.\x1452.\x1453.\x1454.\x1455.\x1456.\x1457.\x1458.\x1459.\x1460.\x1461.\x1462.\x1463.\x1464.\x1465.\x1466.\x1467.\x1468.\x1469.\x1470.\x1471.\x1472.\x1473.\x1474.\x1475.\x1476.\x1477.\x1478.\x1479.\x1480.\x1481.\x1482.\x1483.\x1484.\x1485.\x1486.\x1487.\x1488.\x1489.\x1490.\x1491.\x1492.\x1493.\x1494.\x1495.\x1496.\x1497.\x1498.\x1499.\x1500.\x1501.\x1502.\x1503.\x1504.\x1505.\x1506.\x1507.\x1508.\x1509.\x1510.\x1511.\x1512.\x1513.\x1514.\x1515.\x1516.\x1517.\x1518.\x1519.\x1520.\x1521.\x1522.\x1523.\x1524.\x1525.\x1526.\x1527.\x1528.\x1529.\x1530.\x1531.\x1532.\x1533.\x1534.\x1535.\x1536.\x1537.\x1538.\x1539.\x1540.\x1541.\x1542.\x1543.\x1544.\x1545.\x1546.\x1547.\x1548.\x1549.\x1550.\x1551.\x1552.\x1553.\x1554.\x1555.\x1556.\x1557.\x1558.\x1559.\x1560.\x1561.\x1562.\x1563.\x1564.\x1565.\x1566.\x1567.\x1568.\x1569.\x1570.\x1571.\x1572.\x1573.\x1574.\x1575.\x1576.\x1577.\x1578.\x1579.\x1580.\x1581.\x1582.\x1583.\x1584.\x1585.\x1586.\x1587.\x1588.\x1589.\x1590.\x1591.\x1592.\x1593.\x1594.\x1595.\x1596.\x1597.\x1598.\x1599.\x1600.\x1601.\x1602.\x1603.\x1604.\x1605.\x1606.\x1607.\x1608.\x1609.\x1610.\x1611.\x1612.\x1613.\x1614.\x1615.\x1616.\x1617.\x1618.\x1619.\x1620.\x1621.\x1622.\x1623.\x1624.\x1625.\x1626.\x1627.\x1628.\x1629.\x1630.\x1631.\x1632.\x1633.\x1634.\x1635.\x1636.\x1637.\x1638.\x1639.\x1640.\x1641.\x1642.\x1643.\x1644.\x1645.\x1646.\x1647.\x1648.\x1649.\x1650.\x1651.\x1652.\x1653.\x1654.\x1655.\x1656.\x1657.\x1658.\x1659.\x1660.\x1661.\x1662.\x1663.\x1664.\x1665.\x1666.\x1667.\x1668.\x1669.\x1670.\x1671.\x1672.\x1673.\x1674.\x1675.\x1676.\x1677.\x1678.\x1679.\x1680.\x1681.\x1682.\x1683.\x1684.\x1685.\x1686.\x1687.\x1688.\x1689.\x1690.\x1691.\x1692.\x1693.\x1694.\x1695.\x1696.\x1697.\x1698.\x1699.\x1700.\x1701.\x1702.\x1703.\x1704.\x1705.\x1706.\x1707.\x1708.\x1709.\x1710.\x1711.\x1712.\x1713.\x1714.\x1715.\x1716.\x1717.\x1718.\x1719.\x1720.\x1721.\x1722.\x1723.\x1724.\x1725.\x1726.\x1727.\x1728.\x1729.\x1730.\x1731.\x1732.\x1733.\x1734.\x1735.\x1736.\x1737.\x1738.\x1739.\x1740.\x1741.\x1742.\x1743.\x1744.\x1745.\x1746.\x1747.\x1748.\x1749.\x1750.\x1751.\x1752.\x1753.\x1754.\x1755.\x1756.\x1757.\x1758.\x1759.\x1760.\x1761.\x1762.\x1763.\x1764.\x1765.\x1766.\x1767.\x1768.\x1769.\x1770.\x1771.\x1772.\x1773.\x1774.\x1775.\x1776.\x1777.\x1778.\x1779.\x1780.\x1781.\x1782.\x1783.\x1784.\x1785.\x1786.\x1787.\x1788.\x1789.\x1790.\x1791.\x1792.\x1793.\x1794.\x1795.\x1796.\x1797.\x1798.\x1799.\x1800.\x1801.\x1802.\x1803.\x1804.\x1805.\x1806.\x1807.\x1808.\x1809.\x1810.\x1811.\x1812.\x1813.\x1814.\x1815.\x1816.\x1817.\x1818.\x1819.\x1820.\x1821.\x1822.\x1823.\x1824.\x1825.\x1826.\x1827.\x1828.\x1829.\x1830.\x1831.\x1832.\x1833.\x1834.\x1835.\x1836.\x1837.\x1838.\x1839.\x1840.\x1841.\x1842.\x1843.\x1844.\x1845.\x1846.\x1847.\x1848.\x1849.\x1850.\x1851.\x1852.\x1853.\x1854.\x1855.\x1856.\x1857.\x1858.\x1859.\x1860.\x1861.\x1862.\x1863.\x1864.\x1865.\x1866.\x1867.\x1868.\x1869.\x1870.\x1871.\x1872.\x1873.\x1874.\x1875.\x1876.\x1877.\x1878.\x1879.\x1880.\x1881.\x1882.\x1883.\x1884.\x1885.\x1886.\x1887.\x1888.\x1889.\x1890.\x1891.\x1892.\x1893.\x1894.\x1895.\x1896.\x1897.\x1898.\x1899.\x1900.\x1901.\x1902.\x1903.\x1904.\x1905.\x1906.\x1907.\x1908.\x1909.\x1910.\x1911.\x1912.\x1913.\x1914.\x1915.\x1916.\x1917.\x1918.\x1919.\x1920.\x1921.\x1922.\x1923.\x1924.\x1925.\x1926.\x1927.\x1928.\x1929.\x1930.\x1931.\x1932.\x1933.\x1934.\x1935.\x1936.\x1937.\x1938.\x1939.\x1940.\x1941.\x1942.\x1943.\x1944.\x1945.\x1946.\x1947.\x1948.\x1949.\x1950.\x1951.\x1952.\x1953.\x1954.\x1955.\x1956.\x1957.\x1958.\x1959.\x1960.\x1961.\x1962.\x1963.\x1964.\x1965.\x1966.\x1967.\x1968.\x1969.\x1970.\x1971.\x1972.\x1973.\x1974.\x1975.\x1976.\x1977.\x1978.\x1979.\x1980.\x1981.\x1982.\x1983.\x1984.\x1985.\x1986.\x1987.\x1988.\x1989.\x1990.\x1991.\x1992.\x1993.\x1994.\x1995.\x1996.\x1997.\x1998.\x1999.\x2000.
```

fig.2

*Error in sequential composition, A connected to B, ecc.* vuol dire che il numero di output di A non è lo stesso numero di input di B. Per risolvere il problema dobbiamo utilizzare l'operatore *split composition* ( ; ) che divide il segnale in uscita da *djembe* in due segnali:

```
import("stdfaust.lib");
process = 60, 0.3, 0.4, 1, button("gate") :

```

Compilando il programma possiamo notare molti nuovi elementi nell'interfaccia grafica, questo perché questi elementi sono parte dell'interfaccia grafica di *freeverb*.

### Trigger automatico e costruzione di una mobile App

Possiamo automatizzare la generazione degli impulsi tramite il circuito *ba.pulsen*. Questo dispositivo ha due input, il primo è la dimensione dell'impulso del numero di campioni, nel nostro caso solo un campione. Il secondo è il periodo dell'impulso in campioni. Quindi il periodo è 44100 campioni, dato che la nostra frequenza di campionamento è 44100 Hz avremo un impulso ogni secondo.

```
import("stdfaust.lib");
process = button("gate")*ba.pulsen(1,44100) :
  pm.djembe(60, 0.3, 0.4, 1) <: dm.freeverb_demo;
```

È possibile anche inserire altri generatori di impulsi con altri periodi di tempo in modo da creare degli effetti poliritmici.

```
import("stdfaust.lib");
process = button("gate")*(ba.pulsen(1,44100) +
  ba.pulsen(1,4410*1.3)) :
  pm.djembe(60, 0.3, 0.4, 1) <: dm.freeverb_demo;
```

Adesso cambiamo nome al nostro programma in alto a sinistra e chiamiamolo *djembe.dsp*.

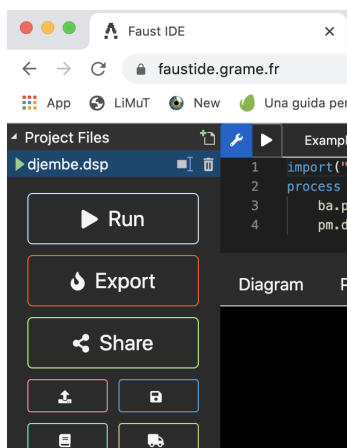


fig.3

Adesso il nostro programma è pronto per essere esportato come un'applicazione Android selezionando *Android platform*

e *Android target* nell'export tool dell'editor online. Purtroppo l'editor online non è in grado di generare direttamente delle applicazioni IOS, ma genera un codice che deve essere successivamente compilato sul tuo PC da terminale.

### Oscillatore a dente di sega

L'obiettivo di questa sezione è costruire un semplice generatore di suoni la cui ampiezza sarà controllata soffiando nel microfono del computer.

Per prima cosa scriviamo un semplice oscillatore a dente di sega con frequenza e gain costanti. Per evitare che il volume sia troppo alto imposteremo il gain a 0.5, tuttavia per poter cambiare facilmente questo parametro in futuro senza modificare la definizione di process possiamo introdurre una nuova variabile chiamata *gain*. Abbiamo due modi di scrivere la nostra linea di process, la prima più intuitiva:

```
import("stdfaust.lib");
import("stdfaust.lib");
gain = 0.5;
//process = os.sawtooth(440)*gain;
//process = gain, os.sawtooth(440) : *;
```

La prima riga di process è una grammatica orientata alla matematica che viene chiamata *infix notation*. La seconda è una scrittura equivalente orientata ai circuiti che consiste nel mettere i dispositivi in parallelo e connetterli in serie ad un moltiplicatore.

Per assegnare un interfaccia grafica al controllo di gain possiamo utilizzare *hslider*, uno slider orizzontale la cui grammatica è: *hslider("nome", valore di partenza, valore minimo, valore massimo, precisione)*

```
import("stdfaust.lib");
gain = hslider("gain",0.5, 0, 1, 0.01);
process = os.sawtooth(440)*gain;
```

### Breath control

Per controllare il nostro oscillatore soffiando nel microfono dobbiamo sostituire lo slider con un envelope follower. In Faust il

dispositivo per il tracciamento dell'ampiezza è *an.amp\_follower\_ar*. Questo circuito ha tre ingressi:

1. Durata dell'attacco dell'involuppo in secondi;
2. Durata del release in secondi;
3. Segnale da analizzare;

```
import("stdfaust.lib");
gain = an.amp_follower_ar(0.02,0.02);
process = gain,os.sawtooth(440):*;
```

### Sintesi additiva

In questa sezione costruiremo un piccolo synth controllato via MIDI. Un piccolo sintetizzatore in sintesi additiva può essere costruito con tre o più oscillatori sinusoidali insieme. Il timbro è determinato dalla relazione dei parametri delle varie armoniche un esempio potrebbe essere il seguente:

```
import("stdfaust.lib");
process = os.osc(440)*0.5 +
  os.osc(440*2)*0.25 +
  os.osc(440*3)*0.125;
```

In Faust l'oscillatore sinusoidale è *os.osc*. Ora dobbiamo generalizzare questo codice per ogni frequenza, per farlo dobbiamo introdurre il concetto di parametro. L'esempio precedente può essere riscritto così:

```
import("stdfaust.lib");
timbro(f) = os.osc(f)*0.5 +
  os.osc(f*2)*0.25 +
  os.osc(f*3)*0.125;
process = timbro(440);
```

In questo modo possiamo generalizzare questa espressione sostituendo la frequenza con un parametro chiamato *f*. Ora possiamo utilizzare la variabile *timbro* all'interno della linea di process.

Ora se vogliamo controllare il gain e la frequenza del timbro dobbiamo creare due nuove interfacce utente chiamate *frequenza* e *gain* utilizzando sempre *hslider*:

```
import("stdfaust.lib");
freq = hslider("freq", 440, 50, 1000, 0.01);
gain = hslider("gain", 0.5, 0, 1, 0.01);
timbro(f) = os.osc(f)*0.5 +
  os.osc(f*2)*0.25 +
  os.osc(f*3)*0.125;
process = gain*timbro(freq);
```

### Sintetizzatore polifonico MIDI

Cerchiamo di capire cosa succede quando premiamo un tasto della nostra tastiera, per prima cosa viene inviato il segnale MIDI *key-on* che contiene due informazioni: il numero del tasto e la velocity. Quando invece lasciamo il tasto viene inviato un messaggio di *key-off* con le stesse informazioni.

In ordine per mappare le informazioni MIDI ai parametri audio del nostro programma dobbiamo rispettare alcune convenzioni per i nomi di questi parametri. Più precisamente devono essere dichiarati tre parametri:

1. Freq = controllato dal *key number*;
2. Gain = controllato dalla *velocity*;
3. Gate = controllato dalla pressione del tasto;

Possiamo quindi testare il programma in due modi differenti: senza polifonia e con il MIDI polifonico. Vediamo un esempio senza nessun device MIDI attaccato implementando un pulsante di gate:

```
import("stdfaust.lib");
freq = hslider("freq", 440, 50, 1000, 0.01);
gain = hslider("gain", 0.5, 0, 1, 0.01);
gate = button("gate");
timbro(f) = os.osc(f)*0.5 +
  os.osc(f*2)*0.25 +
  os.osc(f*3)*0.125;
process = gate*gain*timbro(freq);
```

Per il prossimo step dobbiamo necessariamente lavorare sul browser Google Chrome perché è l'unico browser che implementa il supporto MIDI. Ora possiamo utilizzare o una virtual MIDI keyboard come la *MidiKeys*<sup>5</sup> nel Mac o *VMPK*<sup>6</sup> su Linux e Windows. Per controllare un synth monofonico

<sup>5</sup><https://flit.github.io/projects/midikeys/>

<sup>6</sup><https://vmpk.sourceforge.io>

dobbiamo impostare il menù *Poly Voice* a sinistra dell'IDE, a 1, viceversa se lo vogliamo polifonico dobbiamo impostare il numero di voci che vogliamo implementare.

Per evitare i click che si sentono quando eseguiamo una nota dobbiamo inserire un involuppo collegato all'uscita del gate. In Faust un ADSR è *en.adsr*. Questo dispositivo ha cinque input, rispettivamente:

1. durata dell'attacco in secondi;
2. durata del decay in secondi;
3. valore di sustain;
4. durata del release;

```
import("stdfaust.lib");
freq = hslider("freq", 440, 50, 1000, 0.01);
gain = hslider("gain", 0.5, 0, 1, 0.01);
gate = button("gate") : en.adsr(0.01, 0.01, 0.9, 0.1);
timbro(f) = os.osc(f)*0.5 +
            os.osc(f*2)*0.25 +
            os.osc(f*3)*0.125;
process = gain*gate*timbro(freq);
```

### Aggiungiamo un effetto

Se vogliamo aggiungere un effetto ad un synth di questo tipo non dobbiamo aggiungerlo in serie nella riga di process, infatti in questo modo l'effetto sarà duplicato per ogni nota che andremo a suonare e il nostro programma risulterà molto pesante e poco efficiente. Per evitare questo possiamo utilizzare la funzione *effect* che è simile alla funzione *process* e rappresenta un bus dove è indirizzato tutto il segnale in uscita da process. Tuttavia visto che il riverbero *dm.zita\_light* è stereo dobbiamo rendere anche l'output di process stereo con un'operatore *split*:

```
import("stdfaust.lib");
freq = hslider("freq", 440, 50, 1000, 0.01);
gain = hslider("gain", 0.5, 0, 1, 0.01);
gate = button("gate") : en.adsr(0.01, 0.01, 0.9, 0.1);
timbro(f) = os.osc(f)*0.5 +
            os.osc(f*2)*0.25 +
            os.osc(f*3)*0.125;
process = gain*gate*timbro(freq) <: _._;
effect = dm.zita_light;
```